Contribution to the Theory of Star Formation using Computer Simulations of Turbulent Molecular Clouds

---

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Andreas Bleuler

von

Zollikon ZH

Promotionskomitee

Prof. Dr. Romain Teyssier (Vorsitz)
Prof. Dr. Ben Moore

Zürich, 2015

Andreas Bleuler
Institute for Computational Science
University of Zurich
Winterthurerstrasse 190
CH-8057 Zürich
Switzerland
ableuler@physik.uzh.ch

Contents

Acknowledgements

First of all, I want to thank Romain Teyssier for guiding me throughout my PhD studies. His ability to make complicated things sound easy, his patience and the always open door to his office made my PhD time a very pleasant experience. I also want to thank Ben Moore for hiring me and for reminding us that scientists should go out and make their knowledge accessible for a broader audience. Thanks to Aurel Schneider, Sebastian Elser, Simone Balmelli, Lorenzo de Vittori, Cedric Huwyler, Simon Grimm, Elena Gavagnin and Christian Reinhardt for being great office companions. A special "thank you" to Elena Gavagnin for proofreading large parts of my PhD manuscript. In general, I want to thank everyone at the Institute for Theoretical Physics and the Institute for Computational Science for making this a very nice place to work. Most of all, I want to thank my family for supporting me in everything I have ever done, be it the quest to become a professional ski instructor or a doctor in astrophysics!

Abstract

Stars play a central role in shaping the history of our Galaxy, from its large scale evolution to the formation of planetary systems. Our understanding of their formation inside large clouds of molecular hydrogen is still incomplete. Due to the complex interplay of various physical effects over a vast range of scales, computer simulations have become an indispensable tool for theoretical studies of star formation. Sink particles are routinely applied in these simulations to represent forming stars and thus model processes happening on scales that cannot be resolved in the simulation. I describe a new sink particle algorithm for the astrophysical simulation code RAMSES. As a main novelty it forms sink particles only on the peak locations of well defined gas clumps. The algorithm thus works in tandem with a newly developed structure finding tool named PHEW. PHEW detects overdense regions and their entire substructure in a fully parallel fashion and can be applied in a broad astrophysical or cosmological context. Tests suggest that the pairing with a structure finding tool improves the identification of sites where stars will form through local gravitational collapse. First simulations of star formation in turbulent molecular clouds performed with the new algorithm predict an overall realistic distribution of stellar masses, while the detailed mechanisms that prevent or promote the formation of low mass stars and brown dwarfs needs to be further examined.

Zusammenfassung

Sterne haben eine zentrale Stellung in der Geschichte unserer Galaxie inne. Unser Wissen über die Geburt von Sternen im Inneren riesiger Wolken molekularen Wasserstoffes ist ist bis dato äusserst lückenhaft. Das komplexe Zusammenspiel verschiedenster physikalischer Effekte über viele Grössenordnungen hinweg macht die Entstehung von Sternen zu einem Gebiet, dem sich mit analytischen Methoden nur schwer zu Leibe rücken lässt. Dies hat Computersimulationen zu einem unverzichtbaren Instrument zur theoretischen Untersuchung der Sternentstehung werden lassen. In solchen Simulationen sind sogenannte "Sink-Teilchen" ein zentrales Element. Sie dienen der Modellierung jener Prozesse, die sich auf Grössenordnungen unterhalb der Auflösungsgrenze der Simulation abspielen. Ich beschreibe die Implementierung eines neuen Sink-Teilchen Verfahrens in den astrophysikalischen Simulationscode RAMSES. Als grösste Neuerung gegenüber bisherigen Sink-Teilchen Verfahren erlauben wir die Entstehung neuer Sink-Teilchen nur innerhalb von wohldefinierten Gasklumpen. Unser Sink-Teilchen Algorithmus arbeitet daher im Zusammenspiel mit einem Algorithmus zur Identifizierung von Strukturen innerhalb der Dichteverteilung des Wasserstoffgases. Tests zeigen, dass die vorherige Identifikation von Gasstrukturen eine zuverlässigere Lokalisierung jener Regionen ermöglicht, die letztlich zu einem einzelnen Stern kollabieren werden. Erste Anwendungen unseres neuen Verfahrens zeigen vielversprechende Resultate bezüglich der Massenverteilung der simulierten Sterne, müssen aber die Frage offen lassen, welche physikalischen Prozesse die Entstehung leichter Sterne und brauner Zwerge erschwert respektive begünstigt.

Preface

This manuscript is a summary of the research activities I have been involved in during my time as a PhD student. The smallest common denominator of those activities is the RAMSES code. During the last years, I have not only become a member of the RAMSES user community but also an active developer of the code. Most of my code development is clearly targeting the application of star cluster formation in molecular clouds. However, especially in the design of the structure finding algorithm described in Chapter 5, we have pushed the scope of the implementation far beyond the direct needs in the context of star formation simulations. Here, it was the application of the algorithm as a halo finding tool that motivated the development.

The outline of my manuscript is as follows: the first chapter contains a brief introduction into the field of star formation and molecular cloud physics. A special emphasis is put on the analytic derivation of the relevant mass, length and time scales of gravitational collapse and conditions for stability. These results are of primary importance for the discussion of resolution requirements and sub grid models in computer simulations. Chapter 1 introduces the RAMSES and summarizes the main code modules in the context of this thesis. I furthermore describe, test and discuss some small improvements of the RAMSES code which are very technical and not directly linked to two new code modules that we have developed during the coarse of my PhD thesis. Chapter 3 contains a published paper on the implementation of an improved sink particle algorithm into the RAMSES code. In Chapter 4, I present simulations of star cluster formation in a 1000 $M_\odot$ molecular cloud which have been performed using the new implementation of sink particles. The simulation results are accompanied by an extensive discussion on resolution requirements. Furthermore, we use simulations of individual cores to compare the physically limited but computationally extremely efficient radiative transfer model used to a more exact radiation-hydrodynamics scheme. These results are in preparation phase for publication. Chapter 5 is thematically somewhat separated from the rest of this thesis as it is not primarily dealing with the problem of star formation. Yet it represents a logical progression of the structure finding tool initially developed for the purpose of sink particle formation, to be applied to halo finding in simulations containing dark matter. The chapter consists of a submitted paper on PHEW, a fully MPI-parallel structure finding algorithm capable of locating overdense regions and their substructures on-the-fly in a RAMSES simulation. Finally, in Chapter 6 I present an outlook on future work that could be done.

The material contained in the present thesis which has already been published or submitted for publication by scientific journals is included as individual chapters in the manuscript. Those chapters contain the entire publications including abstract and introduction. Only small changes to make style and layout match the rest of the thesis have been applied.

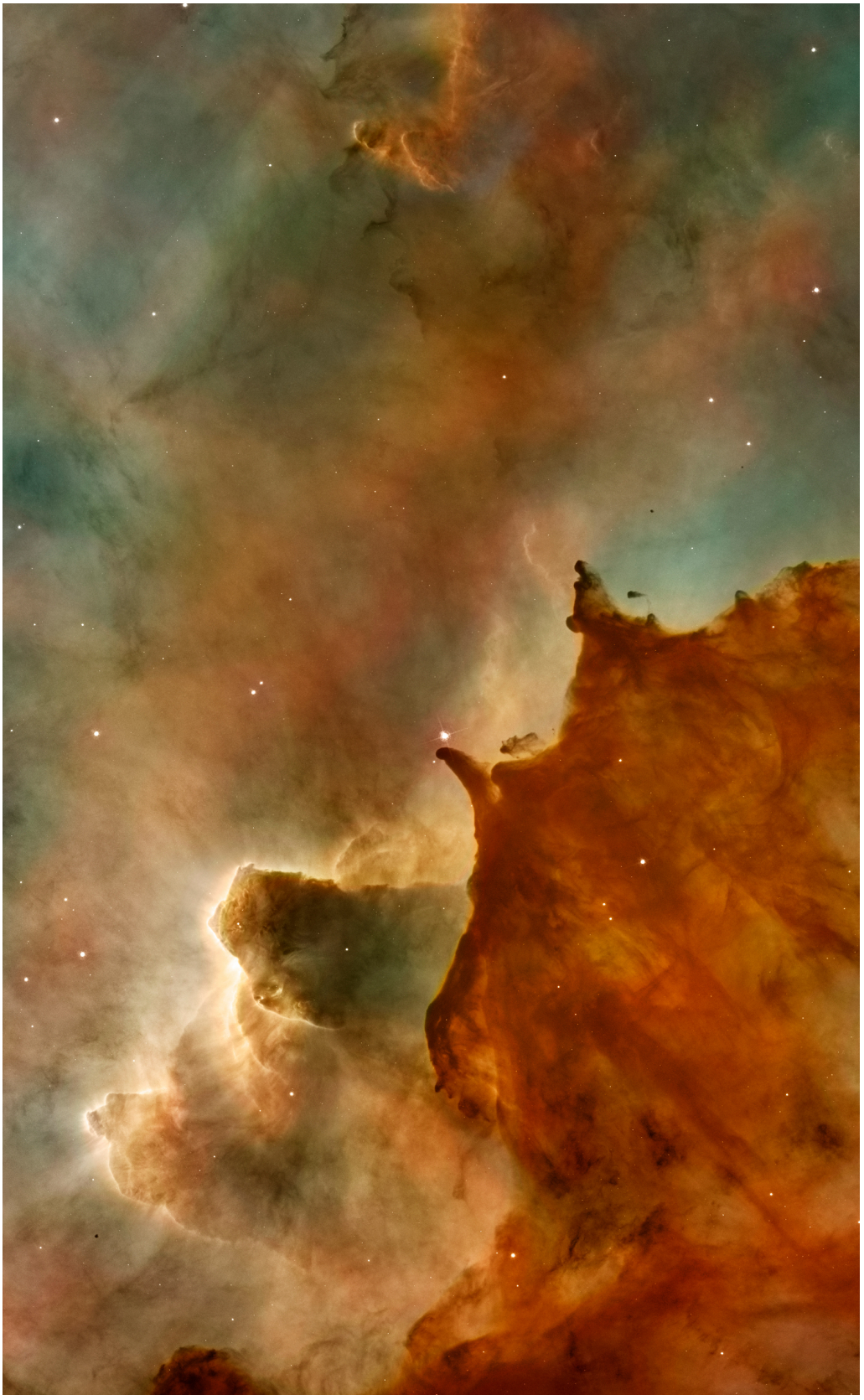


Figure 1: Detail of the Carina Nebula. *Image credit: NASA, ESA, N. Smith (University of California, Berkeley), and the Hubble Heritage Team (STScI/AURA)*

1

Introduction

Some of the most spectacular views ever captured by telescopes are those of star forming molecular clouds. A - in my eyes - particularly beautiful example is shown in Figure 1 where a detail of the Carina nebula is shown[1]. Yet the fact that these images make for such amazing screen savers is also a curse, at least for astronomers trying to understand the physical processes that result in the formation of stars. The wild shapes, the detail at all scales, the contrast between bright regions illuminated by young massive stars and dark sections that entirely block light from their inside, all properties that make molecular clouds so attractive to look at also add to their complexity as study objects. The observed shapes are the result of a multitude of physical effects at play: radiation, chemical processes, magnetic fields, thermal pressure, turbulence and of course gravity are all believed to play a crucial role in the formation of new stars. Star formation must therefore be seen as a *multi-scale* and *multi-physics* problem. As difficult as the task might be, gaining more insight into star formation is very rewarding. During their lifetime and demise, stars release large amounts of energy and heavy elements into interstellar space. This is of great importance for our understanding of a wide range of other processes, from the evolution of entire galaxies to the formation of planetary systems.

In this first chapter, I will give an introduction into the field of star formation that should enable a non-specialized physicist to understand the following chapters of my thesis. I start by very briefly sketching the process of star cluster formation in a molecular cloud. Then I present some analytical approaches to the problem of gravitational stability and collapse. Theses concepts are fundamental for the remainder of the entire manuscript. The results will then help to understand and interpret the most fundamental observational results: the density, velocity and thermal structure of molecular clouds and the observed mass function of stars.

For a more thorough introduction into the field of star formation I refer to the textbooks by Ward-Thompson et al. (2011) and Stahler & Palla (2008). Furthermore, there are many excellent review papers available. Some report on the current status of the entire field in a more condensed (Klessen, 2011) or rather complete (Krumholz, 2014) fashion.

[1] The actual image is a false-color composition of observations carried out at various wavelengths.

Other review articles focus on special aspects of star formation and will be cited at the appropriate locations in this chapter.

1.1. Star formation in a nutshell

Stars form in clusters inside molecular clouds. Molecular clouds are a highly turbulent and clumpy medium. The presence of supersonic turbulence is responsible for the "clumpiness" as it creates regions of compressed gas with densities that can be orders of magnitude above the average cloud density. The clumps themselves are not homogeneous but contain smaller, even denser subclumps. Some clumps are compressed enough to become gravitationally bound. This prevents them from getting dispersed and leads to further contraction. Gravitationally bound clumps inside molecular clouds are believed to result in the formation of a star cluster. The smallest objects that do not contain any gravitationally bound substructure are named *cores* and will presumably give birth to a single star or small stellar systems. A cluster-forming region with its subclumps is shown in the first panel of Figure 1.1. While the entire region undergoes contraction, the cores contract fastest and start to form first protostars. Protostars are dense, pressure supported regions of dissociated molecular hydrogen with a size of a few solar radii. They are still accreting and have thus not yet reached their final mass, however each of them is believed to inevitably result in one individual star or brown dwarf. While protostars are still forming and accreting, their parent cores and clumps might merge which results in multiple protostars accreting from the same gas reservoir, therefore competing for mass (panel 5). In this regime, dynamical interactions of the protostars become more and more important to determine the accretion rate (panel 6). When the gravitational interactions are dominant, low mass protostars might get ejected due to 3-body interactions which terminates their accretion phase (panel 7). Formation of new protostars and accretion onto them continue until the gas is expelled from the cluster, most likely due to feedback effects from massive stars (panel 8). This finally results in a bound or unbound star cluster which is dominated by N-body dynamics (panel 9).

It is fair to say that star formation is not very well understood in general. This is illustrated by the fact that not even in such a brief summary it is possible to avoid some controversial issues. In fact, crucial aspects of three important unresolved questions which were used recently by Krumholz (2014) to build en extensive review, already emerge from the above description. The first one is the problem of the *star formation rate*. Predictions of the amount of gas that will turn into stars in a galaxy within a certain time require - amongst other things - an understanding of the effects that terminate accretion and star formation in a single cluster. Hopeful candidates are ultra-violet radiation pressure from massive stars, stellar outflows or eventually supernovae from the first massive stars that have formed. Closely related is the question of the *spatial and temporal distribution* of star formation. It is generally accepted that stars form in clusters (Lada & Lada, 2003). How is it possible then that the majority of stars in our galaxy is *not* found in clusters anymore? The answer to this question must be closely linked to the processes that remove the gas from the cluster. Finally, the third question is related to the relative importance of the processes depicted in panels 3-5 in Figure 1.1: what determines the *mass distribution* of stars? In particular, to what extent is the final mass of the star already determined at the moment it is born as a protostar by the properties of the parent core? Or is it mainly the competition for gas between accreting protostars that sets the mass? This question is often stated as a *nature or nurture* problem (Bonnell et al., 2004).

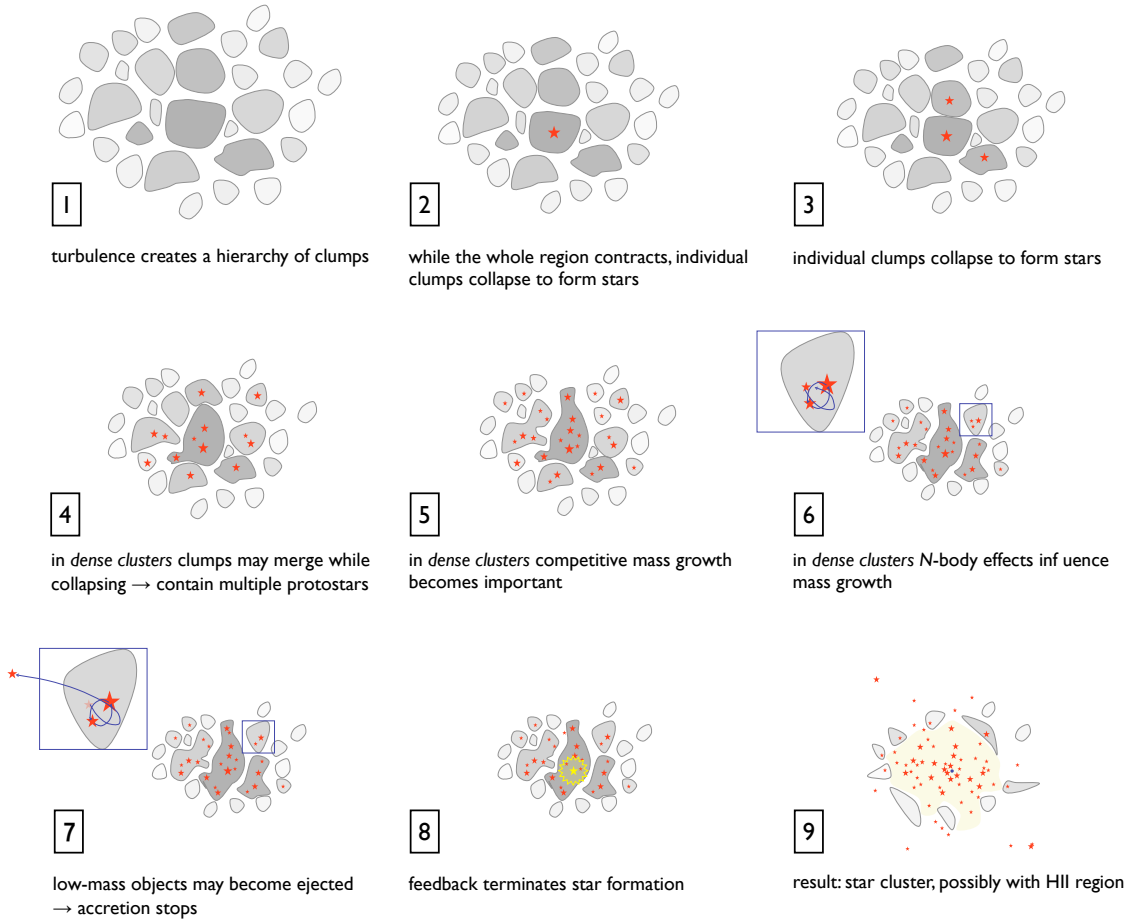


Figure 1.1: Schematic sequence describing the formation of a star cluster in a molecular cloud. Note that the terminology adopted in this figure does not distinguish between clumps in general and what is denoted as a core throughout this work. Image from Klessen (2011), reproduced by permission.

1.2. Gravitational collapse and stability

The brief description given in the previous section has made one thing clear: we need some theoretical foundation on the stability and collapse of self-gravitating gas. However, approaching the physics of molecular clouds with analytic means is a terribly difficult task. On one side, there are various physical effects at play at comparable energy scales. On the other hand, the morphology of molecular clouds is very complex. Consequently, analytic arguments have to rely on crude approximations, such as completely neglecting certain physical effects or assuming more regular geometries like spherical symmetry. Nevertheless, the results obtained in such a way provide a basic understanding of molecular cloud physics. Furthermore, they reveal some relevant scales which are of great importance to define resolution requirements, time constraints and for designing sub grid models for computer simulations. In this section I present three key concepts that deal with gravitational collapse and stability, namely the free-fall time, the Jeans length and the virial theorem.

1.2.1. The free-fall time

A very useful estimate of the relevant time scale for gravitational collapse is the free-fall time. We imagine a homogeneous sphere of perfect pressure-less fluid - usually referred to as dust - with density ρ_0 and radius R, initially at rest. We now compute the time it takes for such a sphere to collapse under its own gravitational attraction. Consider a dust particle at radius r. According to Gauss' law the gravitational field at radius r is that of a point mass in the center of the sphere, containing all the mass inside the radius r,

$$\ddot{r} = -\frac{GM(r)}{r^2}. \tag{1.1}$$

For a homogenous sphere we have $M(r) \propto r^3$. The gravitational acceleration is thus proportional to the position r and the relative acceleration $\ddot{r}/r$ is constant throughout the sphere. Consequently, a homogeneous sphere initially at rest will still be homogeneous after a small time increment Δt and the velocity will be proportional to the position r of the dust particle. The same argument holds thus for later times. As a consequence, the dust sphere is undergoing a uniform contraction, forming a singularity at the center after one so-called *free-fall time* $t_{\rm ff}$. The gravitational acceleration of a particle during the collapse can therefore be written as

$$\ddot{r} = -\frac{4\pi G\rho_0 r_0^3}{3r^2}, \tag{1.2}$$

since the mass inside a particles radius $r(t)$ is constant in time. This problem is equivalent to the problem of a test mass falling onto a point mass

$$M = \frac{4}{3}\pi r_0^3 \rho_0. \tag{1.3}$$

Rather than solving the second order differential Equation 1.2 directly, we use conservation of energy $E_{\rm kin}(t) + E_{\rm pot}(t) = E_{\rm pot}(t = 0)$ to transform the problem into a first order differential equation,

$$\dot{r}^2 = \frac{8\pi^2 G}{3}\Big(\frac{r_0}{r} - 1\Big). \tag{1.4}$$

We take the square root, choose the sign that gives rise to a collapsing rather than expanding solution and apply the substitutions $\xi = r/r_0$ and $\tau = \sqrt{8/3\pi G\rho_0}$ to obtain

$$\frac{\mathrm{d}\xi}{\mathrm{d}\tau} = -\sqrt{\frac{1}{\xi} - 1}. \tag{1.5}$$

Another substitution $\xi = \cos^2\alpha$ yields

$$\frac{\mathrm{d}\alpha}{\mathrm{d}\tau} = \frac{1}{2\cos^2\alpha}. \tag{1.6}$$

Separating the variables and integrating α from 0 to $\pi/2$ (which corresponds to r going from r_0 to 0),

$$\int_0^{\tau_{\mathrm{ff}}} \mathrm{d}\tau = \int_0^{\pi/2} 2\cos^2\alpha\, d\alpha, \tag{1.7}$$

results in the free-fall time,

$$\tau_{\mathrm{ff}} = \frac{\pi}{2} \implies t_{\mathrm{ff}} = \sqrt{\frac{3\pi}{32G\rho_0}}. \tag{1.8}$$

This is the time it takes for a homogeneous sphere of a pressure-less fluid to collapse. Furthermore, it provides us with a general estimate for the relevant dynamical time scale of self-gravitational interactions at a given density ρ_0.[2]

1.2.2. The Jeans length

As a first approach to the problem of gravitational instability in self-gravitating gas, we follow the classical argument of Jeans (1902) who studied small fluctuations in an infinitely extended, homogeneous, self-gravitating gas using linear perturbation analysis. The fundamental equations describing such a system are

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{v}) = 0 \tag{1.9}$$

$$\frac{\partial\boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} = -\frac{c_s^2}{\rho}\nabla\rho - \nabla\phi \tag{1.10}$$

$$\Delta\phi = 4\pi G\rho. \tag{1.11}$$

Note that in Equation 1.10 we have already eliminated the pressure by assuming an isothermal equation of state

$$P = c_s^2\rho, \tag{1.12}$$

where c_s is the isothermal sound speed[3]. This closes the otherwise underdetermined system of Equations 1.9 We now assume a static background configuration given by

$$\rho_0 = const \tag{1.13}$$

$$\boldsymbol{v}_0 = 0. \tag{1.14}$$

$$\tag{1.15}$$

[2]Consequently, the free-fall time it is used as an upper bound for the allowed time step in RAMSES.

[3]The assumption of an isothermal gas at this point is only a mathematical simplification. The adiabatic sound speed of the small perturbations that we will consider later is - to first order - equal to the isothermal sound speed.

Plugging the above relations in Equation 1.10 yields

$$0 = \nabla\phi \implies \Delta\phi = 4\pi G\rho = 0. \tag{1.16}$$

The assumed background state is thus no solution of the set of Equations under consideration. This inherent contradiction is known as the "Jeans swindle". It can be circumvented in various ways, such as the assumption of uniformly rotating background state or that of a (unrealistic) cosmological vacuum energy term which exactly balances the gravitational attraction of the unperturbed background density (Binney & Tremaine, 2011; Shu, 1991). We will ignore this inconsistency as the main conclusions remain valid nevertheless.

We consider small perturbations

$$\rho = \rho_0 + \rho_1,\ \rho_1 \ll \rho_0 \tag{1.17}$$

$$\boldsymbol{v} = \boldsymbol{v}_1,\ |\boldsymbol{v}_1| \ll c_s, \tag{1.18}$$

of the background state. Plugging this in Equations (1.9 - 1.11) and discarding all perturbation terms of quadratic or higher order yields

$$\frac{\partial\rho_1}{\partial t} + \rho_0\nabla\boldsymbol{v}_1 = 0, \tag{1.19}$$

$$\frac{\partial\boldsymbol{v}_1}{\partial t} = -\frac{c_s^2}{\rho_0}\nabla\rho_1 - \nabla\phi, \tag{1.20}$$

$$\Delta\phi = 4\pi G\rho_0. \tag{1.21}$$

We compute the spatial derivative of Equation 1.20 and use Equation 1.21 to eliminate the potential ϕ,

$$\nabla\frac{\partial\boldsymbol{v}_1}{\partial t} = -\frac{c_s^2}{\rho_0}\Delta\rho_1 - 4\pi G\rho_0. \tag{1.22}$$

$$\tag{1.23}$$

After interchanging the spatial and the temporal derivative on the left hand side, we can use Equation 1.19 to eliminate the velocity and obtain

$$\frac{\partial^2\rho_1}{\partial t^2} - c_s^2\Delta\rho_1 = -4\pi G\rho_0^2. \tag{1.24}$$

$$\tag{1.25}$$

This is a regular wave equation with a constant driving term. We restrict the form of the small perturbations to plane waves,

$$\rho_1(\boldsymbol{x}, t) = ae^{i(\boldsymbol{k}\boldsymbol{x}-\omega t)}. \tag{1.26}$$

$$\tag{1.27}$$

Substituting this ansatz in the wave equation leads to a dispersion relation

$$\omega^2 = |\boldsymbol{k}|^2 c_s^2 - 4\pi G\rho_0. \tag{1.28}$$

The right hand side of the above relation can become negative for large background densities ρ_0, which causes an imaginary frequency ω and thus an exponential growth of the amplitude in the plane wave solution 1.26. The corresponding modes are thus unstable and undergo gravitational collapse. For the unstable case, the growth rate of a given mode is

$$\gamma = \sqrt{4\pi G\rho_0 - \frac{4\pi^2 c_s^2}{\lambda^2}}, \tag{1.29}$$

where we have a replaced the wave number using the wave length, $|\boldsymbol{k}| = 2\pi/\lambda$. The perturbations with the largest wavelengths are those with the highest growth rate and thus collapse quickest. By looking at the limiting case $\omega = 0$ we find the shortest wavelength that - if present - will trigger gravitational collapse. We denote this as the Jeans length,

$$\lambda_{\mathrm{J}eans} = \sqrt{\frac{\pi c_s^2}{G\rho_0}}. \tag{1.30}$$

By arguing that inside an object of a given size, there are always perturbations with a wavelength comparable to the objects size itself, we can interpret the Jeans length as the maximum allowed size of gaseous configuration that can be stabilized against gravitational collapse by thermal pressure only. Considering the free-fall time derived in in Section 1.2.1, we can interpret the Jeans length as the scale l above which the free-fall time is shorter than the time it takes for a sound wave to cross the object,

$$\frac{l}{c_s} > t_{\mathrm{ff}} \implies l > \sqrt{\frac{3\pi c_s^2}{32 G\rho_0}}, \tag{1.31}$$

which is the Jeans length up to a factor of $\sqrt{3/32} \approx 1/3$. Using the background density, the length scale given by the Jeans length can be translated into a mass scale. We find the Jeans mass

$$M_{\mathrm{J}eans} = \frac{4\pi}{3}\left(\frac{\lambda_{\mathrm{Jeans}}}{2}\right)^3. \tag{1.32}$$

It is interesting to analyze the the Jeans mass as a function of background density for a polytropic equation of state $p \propto \rho^\gamma$. In such a model, the sound speed is given by $c_s \propto \rho^{\Gamma-1}$ and the thus $M_{\mathrm{J}eans} \propto \rho^{4/3}$. For $\Gamma > 4/3$ the Jeans mass increases with the density. At a certain density, it will be bigger than the entire mass of the collapsing object and the collapse comes to a halt. For $\Gamma < 4/3$, the Jeans mass decreases with increasing density meaning that a contracting object will grow more and more unstable at higher densities. Note that $\Gamma = 4/3$ is exactly the exponent above which stable *polytropes*[4] can exist (Bonnor, 1958).

[4] A polytrope is a spherically symmetric equilibrium solution for a self-gravitating gas sphere where the gas follows a polytropic equation of state.

1.2.3. The virial theorem

The stability analysis in the previous section compares the competing effects of gravity and thermal pressure, completely neglecting any other kind of support. A more complete analysis should include the stabilizing effect of turbulence, radiative pressure and magnetic fields. In this section, we derive the virial theorem including all these additional effects. A very similar but less complete derivation is carried out in Section 3.3.2, where we use a virial relation as a criterion for sink particle formation.

We start by defining the symmetric tensor[5]

$$I_{ij} = \int_{\Omega(t)} \rho r_i r_j \, \mathrm{d}V. \tag{1.33}$$

We now compute the first derivative in time of the tensor I_{ij} in a Lagrangian sense, assuming that the integration volume $\Omega(t)$ is moving and deforming with the flow. By applying Reynolds transport theorem, the divergence theorem and using the continuity equation we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} I_{ij} = \int_{\Omega(t)} \frac{\partial \rho}{\partial t} r_i r_j \, \mathrm{d}V + \int_{\partial\Omega(t)} \rho r_i r_j \boldsymbol{v} \, \mathrm{d}\boldsymbol{S} \tag{1.34}$$

$$= \int_{\Omega(t)} \nabla \cdot (\rho \boldsymbol{v}) r_i r_j + \nabla \cdot (\rho r_i r_j \boldsymbol{v}) \, \mathrm{d}V \tag{1.35}$$

$$= \int_{\Omega(t)} \rho (r_i v_j + r_j v_i) \, \mathrm{d}V. \tag{1.36}$$

We repeat the above step and compute the second derivative in time of I_{ij}

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2} I_{ij} = \frac{1}{2} \int_{\Omega(t)} \frac{\partial \rho}{\partial t}(r_i v_j + r_j v_i) + \rho (r_j \frac{\partial v_i}{\partial t} + r_i \frac{\partial v_j}{\partial t}) + \nabla \cdot \big(\rho (r_i v_j + r_j v_i) \boldsymbol{v}\big) \mathrm{d}V \tag{1.37}$$

$$= \int_{\Omega(t)} \rho v_i v_j + \frac{1}{2} \left(r_i \frac{\mathrm{D}v_j}{\mathrm{D}t} + r_j \frac{\mathrm{D}v_i}{\mathrm{D}t} \right) \mathrm{d}V. \tag{1.38}$$

The first term in Equation 1.38 is twice the kinetic energy tensor $\overline{\overline{T}}$of the gas inside $\Omega(t)$ and contains support due to internal velocities. We split the acceleration into two contributions, one which is due to a density dependent force field $\boldsymbol{a}$ and one which is due to a general stress tensor $\overline{\overline{\sigma}}$,

$$\rho \frac{\mathrm{D}v_i}{\mathrm{D}t} = \rho a_i + \partial_k \sigma_{ki}, \tag{1.39}$$

where we have used the Einstein summation convention. Substituting this in Equation 1.38 we get

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2} I_{ij} = 2T_{ij} + \frac{1}{2} \int_{\Omega(t)} \rho r_i a_j + \rho r_j a_i + r_i \partial_k \sigma_{kj} + r_j \partial_k \sigma_{ki} \, \mathrm{d}V. \tag{1.40}$$

We rewrite the σ-terms,

$$r_i \partial_k \sigma_{kj} = \partial_k (r_i \sigma_{kj}) - \sigma_{ij}, \tag{1.41}$$

[5]The tensor I_{ij} is sometimes referred to as the moment of inertia tensor. We will avoid this to prevent confusion with the tensor $\Theta_{ij} = \int_\Omega \rho(\delta_{ij}|r|^2 - r_i r_j)\mathrm{d}V$ that arises when studying the dynamics of solid body rotations.

which allows to express the gradient of the stress tensor as the sum of a volume and a surface term,

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}I_{ij} = 2T_{ij} + \frac{1}{2}\int_{\Omega(t)} \rho r_i a_j + \rho r_j a_i + \sigma_{ij} + \sigma_{ji}\,\mathrm{d}V + \frac{1}{2}\int_{\partial\Omega(t)} (r_i\sigma_{kj} + r_j\sigma_{ki})n_k\,\mathrm{d}A, \tag{1.42}$$

where $\boldsymbol{n}$ is an outward pointing normal vector to the surface of $\Omega(t)$.

Before specifying the nature of forces and stresses, we specify the frame of reference we are working in: just as the inertia tensor that arises when studying the dynamics of rigid bodies, the tensor I_{ij} depends on the choice of the coordinate system. Starting from the value in the center of mass frame, I^0_{ij} we find $I^{\boldsymbol{s}}_{ij}$, the same value expressed in a coordinate system with center of mass located at position $\boldsymbol{s}$ using the equivalent of the parallel-axis theorem

$$I^{\boldsymbol{s}}_{ij} = \mathcal{I}^{\boldsymbol{0}}_{ij} + M s_i s_j. \tag{1.43}$$

While the above derivations of the generalized virial theorem hold for any inertial frame of reference, the interpretation of I_{ij} as a measure of the size and the shape of the gas configuration inside $\Omega(t)$ only makes sense as long at the coordinate origin is located in its center of mass. We therefore choose the frame which is co-moving with the center of mass of the gas contained in $\Omega(t)$. This non-inertial frame gives rise to a fictitious acceleration that enters Equation 1.39. As long as the frame is non-rotating, this additional acceleration $\boldsymbol{a}^{\mathrm{fict}}$ is independent of the position and its contribution to the second term on the right-hand side of Equation 1.42 vanishes in the center of mass frame,

$$\int_{\Omega(t)} \rho\, r_i a^{\mathrm{fict}}_j\,\mathrm{d}V = a^{\mathrm{fict}}_j \int_{\Omega(t)} \rho r_i\,\mathrm{d}V = a^{\mathrm{fict}}_j s_i \tag{1.44}$$

which vanishes since $\boldsymbol{s} = 0$ in the center of mass frame. We discard the information about the "shape" of the gas and proceed our analysis with the scalar moment of inertia $I = Tr(\overline{\overline{I}})$. Computing the trace of the above expression we obtain

$$\frac{1}{2}\frac{\mathrm{d}^2 I}{\mathrm{d}t^2} = 2T + \int_{\Omega(t)} \rho\,\boldsymbol{r}\cdot\boldsymbol{a} + \mathrm{Tr}(\sigma)\,\mathrm{d}V + \int_{\partial\Omega(t)} (\sigma\cdot\boldsymbol{r})\cdot\boldsymbol{n}\,\mathrm{d}A. \tag{1.45}$$

This is a very general formulation of the virial theorem. In order to derive more familiar versions one has to make further assumptions regarding the nature of forces and stresses. We consider an inviscid fluid which is subject to gravity, magnetic forces and thermal pressure. The momentum equation for such a fluid is given by

$$\rho\frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} = \rho\boldsymbol{g} + \nabla\Big(- \mathbb{1}\Big(P + \frac{B^2}{2}\Big) + \boldsymbol{B}\otimes\boldsymbol{B}\Big). \tag{1.46}$$

Substituting this in Equation 1.45 we obtain

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}^2 I}{\mathrm{d}t^2} = &\int_{\Omega(t)} \rho|\boldsymbol{v}|^2\,\mathrm{d}V + \int_{\Omega(t)} \rho\boldsymbol{r}\cdot\boldsymbol{g}\,\mathrm{d}V + \int_{\Omega(t)} 3P\,\mathrm{d}V - \int_{\partial\Omega(t)} P\,\boldsymbol{r}\cdot\boldsymbol{n}\,\mathrm{d}A \\ &+ \int_{\Omega(t)} \frac{3}{2}|\boldsymbol{B}|^2\,\mathrm{d}V - \int_{\partial\Omega(t)} \frac{|\boldsymbol{B}|^2}{2}\,\boldsymbol{r}\cdot\boldsymbol{n}\,\mathrm{d}A + \int_{\partial\Omega(t)} (\boldsymbol{B}\cdot\boldsymbol{r})(\boldsymbol{B}\cdot\boldsymbol{n})\,\mathrm{d}A.\end{aligned} \tag{1.47}$$

It is worth having a closer look at the terms on the right hand side in the above equation. By assuming that the gravitational potential is entirely due to the matter inside $\Omega(t)$ only, one can write

$$\phi(\boldsymbol{r}) = G \int_{\Omega(t)} \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r}' - \boldsymbol{r}|} \, \mathrm{d}V'. \tag{1.48}$$

This allows the identification of the second term as twice the potential energy. This identification is not valid for gas which is embedded in a complex structure such as a giant molecular cloud with stars as strong external sources of gravity. The third term is the internal energy stored in the three translational degrees of freedom. If one assumes constant pressure inside and on the surface of $\Omega(t)$, the term 3 is cancelled *exactly* by the surface pressure term. This follows directly from Equation 1.46 which contains only the gradient of the pressure. Using the same argument, we conclude that the surface and the volume terms for the magnetic fields cancel as well in the case of a homogeneous external magnetic field penetrating the gas inside $\Omega(t)$. A large scale galactic field does therefore not directly stabilize a molecular cloud against collapse. Only variations of the magnetic field on the scale of the cloud itself do add support to the cloud. For objects with a very large density and pressure contrast against the surrounding medium, monoatomic gas and small scale fluctuations dominating the magnetic field, we can safely ignore the surface terms and we recover the very well know form of the virial theorem

$$2E_{\mathrm{kin}} + 2E_{\mathrm{therm}} + E_{\mathrm{grav}} + E_{\mathrm{mag}} = 0, \tag{1.49}$$

as a condition for stability. To recover a Jeans-like criterion, we consider an isolated homogeneous sphere of radius R, supported by pressure only. The virial theorem then reads

$$8\pi c_s^2 \rho R^3 - \frac{16}{15}\pi^2 \rho^2 G R^5 = 0 \tag{1.50}$$

which leads to

$$R = \sqrt{\frac{15 c_s^2}{2G\rho}}. \tag{1.51}$$

Increasing the size lets the gravitational energy grow as R^5 while the pressure term grows as R^3. Using the virial theorem, we have thus obtained a very similar criterion for the maximum mass of an object that can be supported by thermal pressure as we have already obtained using the Jeans argument.

1.2.3.1. The sound scale

To draw another consequence of the virial theorem, we compare the support by internal velocities to pressure support. Again neglecting the surface pressure term and assuming an isothermal equation of state $P = c_s \rho$ we equate the two types of support,

$$\int_{\Omega(t)} \rho |\boldsymbol{v}|^2 \, \mathrm{d}V = \int_{\Omega(t)} 3 c_s^2 \rho \, \mathrm{d}V. \tag{1.52}$$

In the chosen frame, the mean velocity vanishes. We can thus express the left hand side of the above equation in terms of the velocity dispersion σ_v and integrate the right hand side,

$$M\sigma_v^2 = 3Mc_s^2 \implies \sigma_v = \sqrt{3} c_s. \tag{1.53}$$

Table 1.1: Physical parameters of clouds, clumps and cores. Numerical values taken from Bergin & Tafalla (2007); Klessen (2011)

	clouds	clumps	cores
Size [pc]	$2-20$	$0.1-2$	$\lesssim 0.1$
Mean density [$H_2\,cm^{-3}$]	10^2-10^3	10^3-10^5	$>10^5$
Mass [$M_\odot$]	10^2-10^6	$10-10^3$	$0.1-10$
Temperature [K]	$10-30$	$10-20$	$7-12$
Line width [$km\,s^{-1}$]	$1-10$	$0.5-3$	$0.2-0.5$
RMS Mach number	$5-50$	$2-15$	$0-2$
Column density [$g\,cm^{-2}$]	0.03	$0.03-1.0$	$0.3-3$
Turbulent crossing time [Myr]	$2-10$	$\lesssim 1$	$0.1-0.5$
Free-fall time [Myr]	$0.3-3$	$0.1-1$	$\lesssim 0.1$

The velocity dispersion typically increases monotonously with the size of the region under consideration (see Section 1.3.1). We can therefore define the *sound scale* $\lambda_{\rm sound}$ as the scale above which velocity dispersions are supersonic,

$$\sigma_v(\lambda_{\rm sound}) = c_s. \tag{1.54}$$

Comparing this with Equation 1.53 shows that at this scale, support due to internal velocities and pressure are of the same order. While the velocity term is dominant at larger scales, pressure provides the main support at smaller scales.

1.3. Molecular cloud properties

In this section I summarize the most important properties of molecular clouds. For a comprehensive review on observational and theoretical results, see Hennebelle & Falgarone (2012). As already mentioned earlier in this chapter, molecular clouds are turbulent, clumpy assemblies of molecular hydrogen, helium and dust, embedded in more diffuse regions of atomic hydrogen in interstellar space. The terminology used to denote molecular clouds and their substructure is not always consistent. Throughout this work we will use the term *cloud* for the entire connected regions where hydrogen is found mostly in molecular phase. The word *clump* is used for pretty much any substructure inside molecular cloud. As *core* we denote the smallest gravitationally bound structure inside molecular clouds. Another defining property of the cores is the sub- or only mildly supersonic velocity dispersion. This is roughly consistent with our "definition" of a core as supersonic velocity fluctuations inside cores would likely result in even more compressed and therefore bound subregions. Ranges of the physical parameters for clouds, clumps and cores are listed in Table 1.1.

1.3.1. Larsons scaling relations

The most fundamental results regarding the density and velocity structure of molecular clouds where presented by Larson (1981) who examined the available observational data on

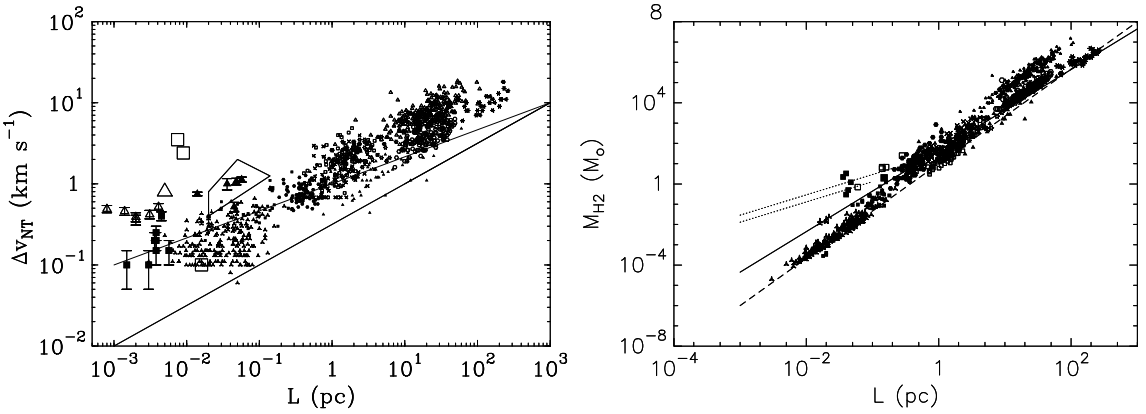


Figure 1.2: Combined data from various molecular cloud surveys examined for the velocity dispersion-size (*left panel*) and mass-size (*right panel*) relation. Figure taken from Hennebelle & Falgarone (2012), reproduced by permission.

molecular clouds for correlations between the cloud masses, sizes and velocity dispersions. His dataset contained entire clouds as well as substructure residing inside clouds, covering a range of approximately $(0.1\text{pc}, 100\text{pc})$ and $(1\text{M}_\odot, 10^5\text{M}_\odot)$ in size and mass respectively. The study resulted in three scaling relations from then on known as Larson's laws. Larson's first law is a velocity dispersion-size relation, $\sigma_v \propto L^{0.38}$. The exponent 0.38 is very close to 1/3 which is the value describing velocity fluctuations in incompressible turbulence (Kolmogorov, 1941a,b). This led Larson to the conclusion that the velocities in molecular clouds are due to large-scale sub- or mildly supersonic turbulence in the surrounding diffuse interstellar medium. The second law is a velocity dispersion-mass relation, $\sigma_v \propto M^{0.20}$. Combining this with the first relation and assuming spherical symmetry yields a good correlation of kinetic and gravitational energy, which Larson interpreted as a hint for the clouds being roughly in virial equilibrium. The third of Larson's laws is a density-size relation, $\rho \propto L^{-1.1}$. This correlation is often expressed in terms of an approximately constant surface density Σ.

Larson's scaling laws have been reexamined in many observations, targeting different objects in the Milky Way or nearby galaxies and using different observational techniques. For an overview of those results see the review by Hennebelle & Falgarone (2012) from where I have taken Figure 1.2 which shows a collection of data points from various surveys for the velocity dispersion-size and the mass-size relationship. Arguably the most commonly accepted (and easy to remember) form of Larsons's laws is

$$\sigma_v \propto \sqrt{L}, \tag{1.55}$$

$$\Sigma = const. \tag{1.56}$$

The exponent in the relation 1.55 is now the one expected for velocity fluctuations in supersonic turbulence. Only the first and the third of Larson's laws are stated here as the three laws are algebraically linked. Any of the three relations can be recovered from the other two. The second law derived from the two above relations is $\sigma_v \propto M^{1/4}$. Combining this with the velocity dispersion-size relation 1.55 in the same way as for the original Larson laws yields $E_{\text{kin}} \propto E_{\text{grav}}$, where the normalization is consistent with virial equilibrium. This complicates the interpretation of Larson's first law 1.55. A turbulent as well as a gravitational origin of the internal velocities are consistent with the observational data.

This picture changes if one allows for variation in the surface density. In this case, $E_{\rm kin} \propto E_{grav}$ implies that $\sigma_v \propto \sqrt{\Sigma L}$. Such variations in the surface density and the corresponding deviation from relation 1.55 are reported in Heyer et al. (2009) and Roman-Duval et al. (2010) who find $M \propto L^{2.36}$ and $\sigma_v \propto \sqrt{\Sigma L}$. Heyer et al. (2009) take this as an argument against a turbulent origin of the velocity dispersion. Kritsuk et al. (2013) on the other hand show that for scales above the sonic length, the $M \propto L^{2.36}$ can be derived from turbulent arguments. Furthermore, they question the variation of the velocity dispersion with the surface density reported by Heyer et al. (2009).

The exact form and interpretation of Larson's laws are thus still heavily debated. In the context of the simulations described in this thesis, Larson's laws are mainly used to create initial conditions for the collapse of a $1000\,\mathrm{M}_\odot$ molecular cloud. For that purpose, we have adopted the mass-size relation described by Roman-Duval et al. (2010) and $\sigma_v(r) \propto \sqrt{r}$ for the internal velocity dispersion as a function of size. The normalization of the velocity dispersion is chosen such that $E_{\rm kin} = \frac{1}{2} E_{\rm grav}$ for the entire cloud. Finally, when combined with the assumption of a constant sound speed, Larson's first law 1.55 leads to a universal sound scale (see Equation 1.54) of $\sim 0.1\,\mathrm{pc}$ in molecular clouds. This is important as we use the sound scale to define a minimum resolution requirement in our simulations.

1.3.2. Thermal structure and the first hydrostatic core

The gravitational stability analysis performed in Sections 1.2.2 and 1.2.3 shows the direct influence of the temperature on the characteristic fragmentation scale. On one hand, the sound speed determines the sound scale defined in Equation 1.54 which is the relevant scale for the formation of cores. On the other hand, the the Jeans length is a function of the sound speed and thus of the gas temperature, $\lambda_{Jeans} \propto c_s \propto \sqrt{T}$. This suggests that the temperature is controlling the fragmentation of cores into individual protostars. We thus need to take a look at the thermal structure of molecular clouds. For more information see for example Bergin & Tafalla (2007).

The temperature of molecular clouds is the result various competing heating and cooling effects. The dominant heating contribution stems from ionization by cosmic rays while the main cooling mechanism is emissions corresponding to changes in the excitation of the rotational state of CO molecules Bergin & Tafalla (2007). This results in a gas temperature of ~ 10 K. At higher number densities ($n \gtrsim 10^4 - 10^5 \mathrm{cm}^{-3}$) collisions between gas molecules and dust grains become frequent enough for the gas and dust temperatures to equalize. At these densities and temperatures the CO begins to "freeze out" onto the dust grains, therefore loosing their role as main coolant. It is now the thermal emission from dust grains which acts as major cooling mechanism. Despite this transition, the gas temperature remains at ~ 10 K. This justifies the assumption of isothermal 10 K gas over a large range of densities that is often employed in simulations.

The picture dramatically changes inside a collapsing region at densities $\sim 10^{-13} \mathrm{g\,cm}^{-3}$. This is the density where the mean free path of dust-emitted infrared photon $\lambda_{\rm IR} = (\kappa_{\rm dust}\rho)^{-1}$ is comparable to the size of the collapsing region, $L \propto M^{1/3}\rho^{-1/3}$. The now well-coupled gas-dust fluid becomes opaque to its own thermal radiation and temperature increases due to compressional heating. The fact that the typical dust opacity increases as T^2 amplifies this effect. Simply put, the gas transitions from an isothermal phase to an adiabatic phase at $\sim 10^{-13} \mathrm{g\,cm}^{-3}$. Gas undergoing contraction in this adiabatic phase is denoted as the first hydrostatic core or the first Larson core. The regime of the first hydrostatic core ends when temperatures are high enough to dissociate hydrogen at $\gtrsim 1000$

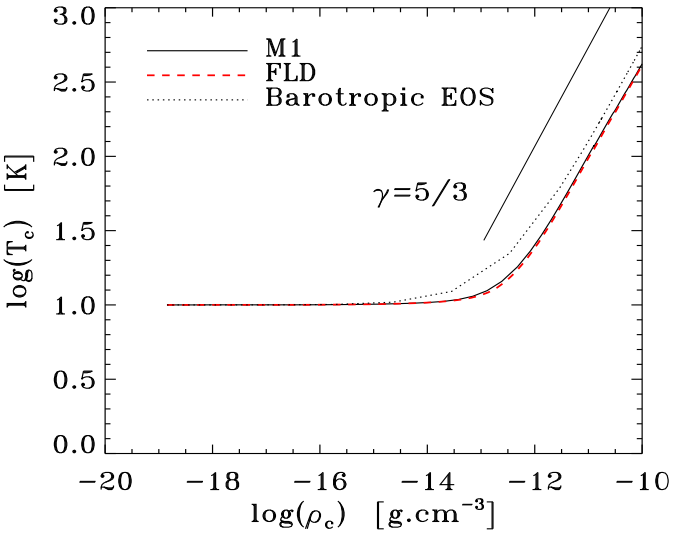


Figure 1.3: Density-temperature diagram resulting from a 1D spherical collapse simulation performed by Commerçon et al. (2011a). Figure reproduced by permission of the author.

K which initiates another phase of (near) free-fall collapse. A quantitative description of these processes was mainly established by 1D radiation hydrodynamical (RHD) simulations assuming spherically symmetric collapse (Commerçon et al., 2011a; Masunaga & Inutsuka, 2000; Masunaga et al., 1998). Figure 1.3 taken from Commerçon et al. (2011a) shows the temperature-density relation obtained from two such calculations using different schemes for the radiative transport and compares it to a barotropic equation of state. Such an equation of state is often used in 3D calculations of fragmenting molecular clouds to mimic the effect of the full (but computationally expensive) radiative transfer. Looking at Figure 1.3, this looks like a justified approximation. However, there is one fundamental difference: the increased temperature in the RHD calculation is controlled by the contraction rate which can be expressed as the rate of change in the density. In the barotropic model it is simply a function of the density. Consequently, the barotropic model can form hydrostatic configurations where contraction is completely stopped. This is clearly unphysical as a vanishing contraction rate would also result in a vanishing heating term while the gas would still cool and therefore loose pressure support.

1.4. The initial mass function

Arguably the most fundamental observational constraint on theories of star formation is the distribution of stellar masses. To account for the different lifetimes of stars with different masses, this distribution is usually quantified as the initial mass function (IMF). If normalized to one, it gives the probability density function of the mass at which a star enters the main sequence. The IMF is constructed by fitting an assumed functional shape to observational data. The first IMF was derived by Salpeter (1955) who fitted his observations in the mass range $M_\star \in [0.4\mathrm{M}_\odot, 10.0\mathrm{M}_\odot]$ with a power law. Nowadays, the two most frequently used functional forms of the IMF are the ones established by Kroupa (2002) and Chabrier (2003, 2005). While Kroupa uses three power laws in three different mass ranges, Chabrier fits observations with a lognormal for the low and medium mass

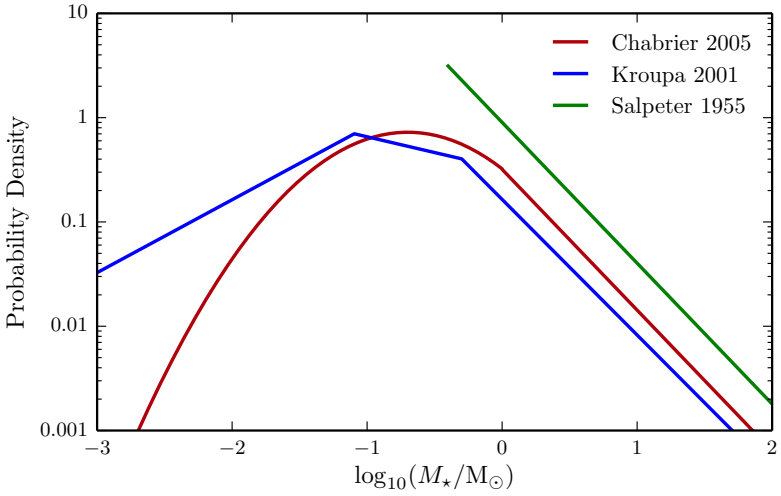


Figure 1.4: Probability density functions for the mass at which a star enters the main sequence after Salpeter, Kroupa and Chabrier IMF. Note that the Salpeter IMF is normalized to one in the mass range $(0.4\mathrm{M}_\odot, \infty)$.

range and switches to a Salpeter-like power law for the high mass tail (see Figure 1.4). Given in terms of the logarithmic mass $x = \log_{10}(M_\star/\mathrm{M}_\odot)$, the IMFs by Salpeter, Kroupa and Chabrier are

$$\xi_{\mathrm{Salpeter}}(x) \propto x^{-1.35}, \tag{1.57}$$

$$\xi_{\mathrm{Kroupa}}(x) \propto \begin{cases} x^{0.7} & \text{if } x < \log_{10} 0.08, \\ x^{-0.3} & \text{if } \log_{10} 0.08 < x < \log_{10} 0.5, \\ x^{-1.3} & \text{if } \log_{10} 0.5 < x, \end{cases} \tag{1.58}$$

$$\xi_{\mathrm{Chabrier}}(x) \propto \begin{cases} \exp\left(-\frac{(x-\log_{10} 0.2)^2}{2\times 0.55^2}\right) & \text{if } x < 0, \\ x^{-1.35} & \text{if } x > 0. \end{cases} \tag{1.59}$$

Observations find the IMF to be relatively invariant throughout the Milky Way and nearby galaxies, where the exact degree of variation in the IMF is subject to lively debates (e.g., Bastian et al., 2010; Offner et al., 2013). Due to its universality, the IMF is usually the first touchstone for any attempt to explain star formation. A theory of star formation must be able to predict the observed IMF and a numerical model has to reproduce it to be considered realistic. There have been various attempts to derive the IMF by analytic means. Here, I briefly introduce two approaches to the problem. The first one is usually denoted as *gravo-turbulent fragmentation* model and adopts the "nature" point of view, whereas the *competitive accretion* model assumes "nurture" to determine stellar masses. For a more detailed overview over analytic derivations of the IMF see Krumholz (2014); Offner et al. (2013).

The starting point for the gravo-turbulent fragmentation model (Hennebelle & Chabrier, 2008; Hopkins, 2012) is the density fluctuations produced by turbulence. The logarithmic density is described as a Gaussian random field with given power spectrum, usually assumed to be $P(k) \propto k^{-4}$. In analogy to cosmology, gravitationally bound overdense regions are identified using the (extended) Press-Schechter formalism (Bond et al., 1991; Press & Schechter, 1974). The smallest gravitationally bound structures that con-

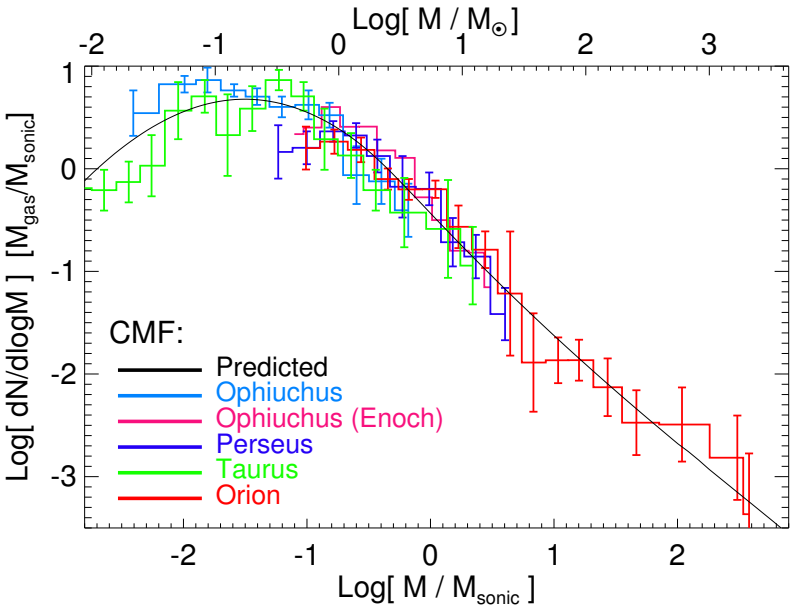


Figure 1.5: Comparison of the predicted last-crossing mass function and the observed core mass functions in various molecular clouds. *Figure from Hopkins (2012), reproduced by permission.*

tain no substructures which are themselves gravitationally bound are identified as cores. The derived mass distribution matches the core mass distribution found in observations (see Figure 1.5). This model is very appealing since it has only two free parameters: the turbulent Mach number and the assumed power spectrum of the logarithmic density fluctuations. The main open question in this model is the connection between the core mass function and the IMF. Their observed shapes are similar with the core mass function being shifted to higher masses by a factor of $\approx 3-4$. However, the straightforward interpretation that one core forms one star with a star formation efficiency of $\approx 25\%$ conflicts with large number of observed binary systems. In general it remains unclear how stars form inside cores, thereby inheriting the shape of the core mass function.

Competitive accretion models assume that accretion sets the mass of a star after it is born (see Bonnell et al. (2007) for a review on those models). The basic assumption is that the accretion rate of a protostar is a monotonously increasing function of its mass, $\dot{m} = f(m)$. A small initial spread in protostellar masses will thus amplify and naturally lead to a power-law like mass function. The slope of the resulting mass function depends on the exact shape of the function f. This is a compelling argument to explain the high mass tail in the IMF. The main problem of the competitive accretion models is that they do not introduce a characteristic mass. The model must therefore be extended to explain the turnover in the IMF. Bonnell et al. (2007) suggest that the characteristic mass might be set by fragmentation and that ejections can stop accretion of some objects early and hence shape the low mass tail.

2

THE RAMSES CODE

RAMSES is an acronym that stands for "Raffinement Adaptatif de Mailles Sans Efforts Surhumains"[1], a name that does not only reveal the nationality of its inventor and main developer (Teyssier, 2002), but also the fact that RAMSES is an adaptive mesh refinement (AMR) code which is relatively easy to use. While RAMSES was originally developed to study the co-evolution of dark matter and gas under self-gravity in a cosmological context, it has evolved into a general tool for simulations of self-gravitating flows with a variety of modules incorporating further physical effects. RAMSES is parallelized using the MPI library[2] and capable of running on thousands of CPUs on state-of-the-art high performance computing infrastructure. RAMSES is an open-source project. The code including the developments described in this thesis can be downloaded from the RAMSES repository located at `http://www.bitbucket.org/rteyssie/ramses`.

In this chapter I briefly introduce some of the key techniques implemented in the RAMSES code. I omit the code modules that have not been used for the simulations resulting in the present thesis. A special emphasis is laid on small code improvements I have worked on, which are too technical to be mentioned in a publication focusing on simulation results, but not important enough to deserve an entire scientific paper on their own.

[1] Loosely translated: "Adaptive mesh refinement for non-superheroes"
[2] `http://www.mpi-forum.org`

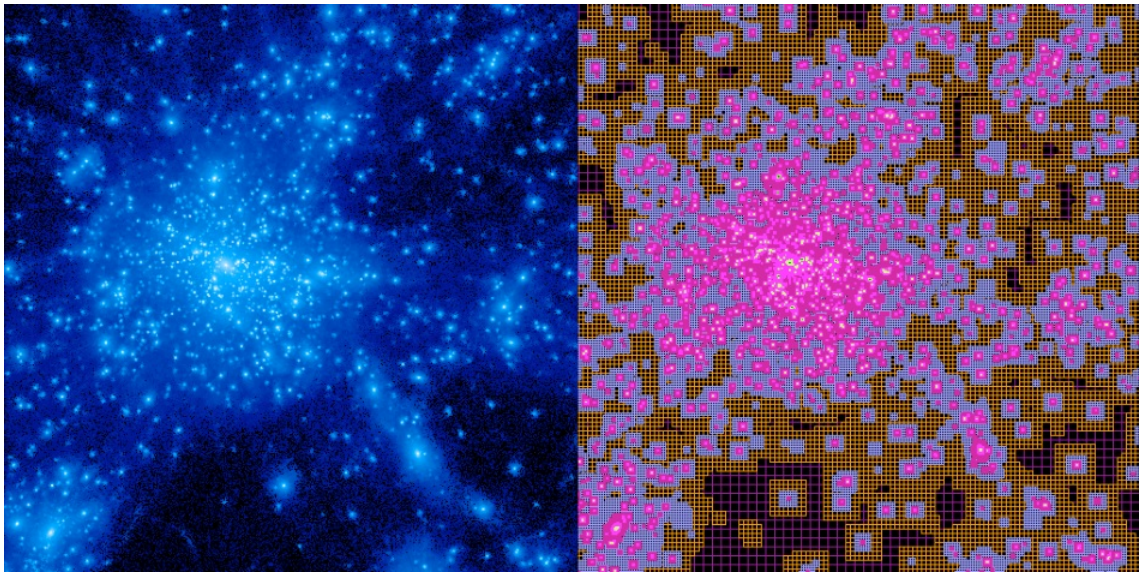


Figure 2.1: RAMSES density distribution together with the corresponding AMR grid. Image from Teyssier/Agertz lecture notes.

2.1. Adaptive Mesh Refinement

The two main techniques for solving the hydrodynamical equations in an astrophysical context are the finite volume method (e.g., LeVeque, 2002; Toro, 2009) and the smoothed-particle hydrodynamics (SPH) method introduced by Gingold & Monaghan (1977); Lucy (1977). Theses two approaches differ fundamentally in the way the fluid is discretized. Finite volume codes discretize the computational domain into volume elements (cells), whereas SPH codes discretize the fluid into mass elements (particles). The volume elements are fixed in space and thus employ an Eulerian approach to hydrodynamics, while the mass elements in SPH are advected with the flow, therefore representing the Lagrangian formulation of hydrodynamics. As a result of that, SPH codes naturally increase the spatial resolution in high density regions[3]. This is very beneficial for simulations of self-gravitating fluids which tend to produce huge density contrasts. A way for finite volume codes to achieve a similar adaptivity in spatial resolution is the adaptive mesh refinement (AMR, Berger & Colella (1989a,b)) technique. AMR simulations allow the definition of criteria which will trigger the partition of a cell into smaller cells (refinement), resulting in a locally increased resolution. Recursively repeating this procedure allows to reach a dynamic range which is far from possible with a fixed cartesian grid. Figure 2.1 shows the projection of a RAMSES density field together with the corresponding AMR grid hierarchy. For a thorough comparison of grid-based and particle-based hydrodynamics in astrophysics and cosmology see Agertz et al. (2007).

There are various ways to implement AMR in practice. The RAMSES data structure is a "fully threaded tree" (Khokhlov, 1998). The main elements in this data structure are *octs* which consist of 2^{ndim} cells. The octs are organized in linked lists, one for each AMR refinement level. Every oct carries pointers to its father cell and the $2 \times ndim$ neighboring

[3]In SPH, hydrodynamical quantities at a particles location are computed by averaging over the nearest N neighbors. A higher number density of SPH particles therefore reduces the radius which contains N neighbors and thus increases the spatial resolution.

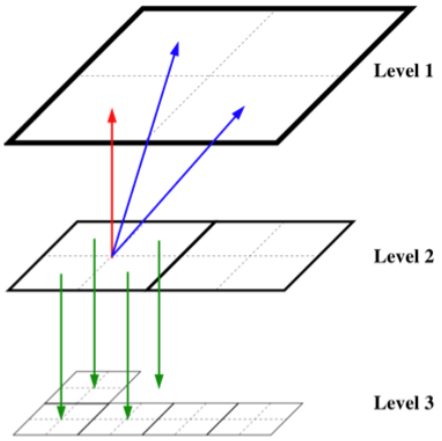


Figure 2.2: Two dimensional representation of the fully threaded AMR tree. Figure from Teyssier/Agertz lecture notes.

father cells. Beyond that, each cell in an oct at level ℓ carries pointers to its children oct at level $\ell + 1$. This structure is shown in Figure 2.2. Other AMR implementations such as patch-based or block-based AMR codes (e.g. ENZO O'Shea et al. (2004), FLASH Fryxell et al. (2000)) refine larger areas at once. The presence of bigger blocks of cells at a given refinement level can be beneficial as they for instance allow designing higher order schemes which require presence of more than just the nearest neighbors to compute derivatives. The fully threaded tree employed in RAMSES on the other hand allows the refinements to exactly trace interesting regions in the flow without wasting resources on adjacent, less important parts of the computational domain. This is important for tracing detailed filamentary or sheet-like structures as they are ubiquitous in simulations of self-gravitating supersonic flows.

2.2. The Hydrodynamics Solver

In this section, I will briefly introduce the key concepts of the numerical scheme to solve the hydrodynamical equations in RAMSES. For an extensive introduction into numerical methods for fluid dynamics I refer to the book by Toro (2009).

It is convenient to reformulate the Euler equations 1.10 in conservative form. Adding the energy equation and neglecting gravitational accelerations they read

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{2.1}$$

$$\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v} + P \mathbb{1}) = 0 \tag{2.2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (E \boldsymbol{v} + P \boldsymbol{v}) = 0, \tag{2.3}$$

where E is the total energy density of the fluid,

$$E = \rho \Big(\frac{\boldsymbol{v} \cdot \boldsymbol{v}}{2} + \epsilon \Big). \tag{2.4}$$

The specific internal energy ϵ is related to the pressure by an ideal gas equation of state

$$P = \rho \epsilon (\gamma - 1) \tag{2.5}$$

which closes the system, γ is the adiabatic index of the gas. The set of equations can be stated in a more compact form in terms of a state vector $\boldsymbol{U} = (\rho, \rho\boldsymbol{v}, E)^T$ and corresponding fluxes $\boldsymbol{F} = (\rho\boldsymbol{v}, \rho\boldsymbol{v} \otimes \boldsymbol{v} + P, (E + P)\boldsymbol{v})^T$,

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F} = 0. \tag{2.6}$$

We have thus expressed the Euler equations as a general conservation law. For simplicity, we now restrict ourselves to the case of only one spatial dimension[4] denoted as x. The above equation becomes

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = 0. \tag{2.7}$$

The finite volume approach to such a conservation law consists in spatially averaging the above equation over *cells* of length Δx. The integral over the derivative of the fluxes is expressed through the fluxes at the cell boundaries. For the cell averaged quantities U_i we obtain

$$\frac{\partial \boldsymbol{U}_i}{\partial t} + \frac{\boldsymbol{F}_{i+1/2} - \boldsymbol{F}_{i-1/2}}{\Delta x} = 0, \tag{2.8}$$

where $\boldsymbol{F}_{i\pm1/2}$ denotes the flux in positive x-direction through the interface between cell i and cell $i \pm 1$. A nice property of the finite-volume approach is the conservation of mass, momentum and energy: the same amount of a conserved quantity that leaves a cell through a given interface enters the adjacent cell. The problem of solving the hydrodynamics equations in a finite-volume scheme thus reduces to computing fluxes through cell interfaces. Integrating the above equation over a finite time step Δt yields the temporal evolution of the cell averages,

$$\boldsymbol{U}_i^{n+1} - \boldsymbol{U}_i^n + \int_{t_n}^{t_n+\Delta t} \frac{\boldsymbol{F}_{i+1/2} - \boldsymbol{F}_{i-1/2}}{\Delta x}\, \mathrm{d}t = 0. \tag{2.9}$$

In the basic Godunov scheme (Godunov, 1959), one assumes $\boldsymbol{U} \equiv \boldsymbol{U}_i$ to be constant throughout the cells. Computing the resulting fluxes at the cell interfaces is thus equivalent to solving the classical *Riemann problem*: a conservation law with piecewise constant initial values, separated by a discontinuity. The self-similarity of the solution to the Riemann problem implies that the resulting flux through the interface I is constant in time as long the time step Δt is chosen small enough to ensure that no waves originating from other cell interfaces can reach I within Δt. Equation 2.9 can thus be written as

$$\frac{\boldsymbol{U}_i^{n+1} - \boldsymbol{U}_i^n}{\Delta t} + \frac{\mathcal{RP}[U_i^n, U_{i+1}^n] - \mathcal{RP}[U_{i-1}^n, U_i^n]}{\Delta x} = 0, \tag{2.10}$$

where $\mathcal{RP}[U_i^n, U_{i+1}^n]$ is the flux between cell i and cell $i + 1$ that results from solving the Riemann problem. For methods to solve the Riemann problem numerically, see Toro (2009)[5].

2.2.1. Slope limiters

To improve the accuracy of the basic Godunov method to second order in space and time, RAMSES uses the MUSCL (**M**onotonic **U**pstream-centered **S**cheme for **C**onservation

[4]The generalization to multiple dimensions in space in *non-trivial*. However, the basic ideas remain valid nevertheless.

[5]The calculations presented in this thesis have been performed using the HLLC Riemann solver.

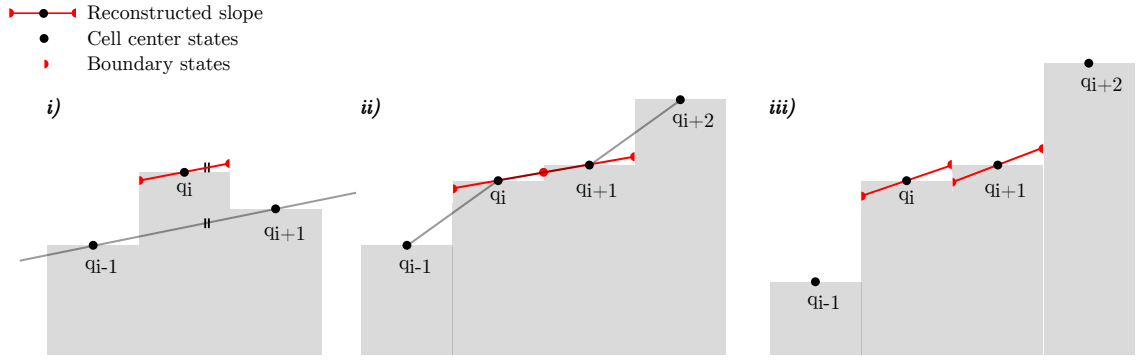


Figure 2.3: Construction of cell interface states using different slopes. In panel i), the unlimited central slope is used. Panel ii) shows the MinMod slope which assures monotonicity of the reconstructed solution across cell boundaries. In panel iii) the MonCen slope is depicted. Here, the slope is limited such that the interface value is bounded by the average value in the adjacent cell.

Laws) scheme introduced by van Leer (1979). The main modification compared to the basic Godunov scheme 2.10 consists in the linear reconstruction of the cell interface states from the cell-averaged quantities. These interface states are then used as input states for the Riemann solver[6]. In RAMSES, the reconstruction of the interface states is performed using the *primitive variables* which are density, velocity and pressure . A component q of the state vector $\boldsymbol{Q}$ of primitive variables is approximated with a piecewise linear function. To compute the slope m of q inside a cell from cell-averaged values q_{i-1}, q_i, q_{i+1}, we consider the following approximations,

$$m_C = \frac{q_{i+1} - q_{i-1}}{2\Delta x} \qquad \frac{\partial q}{\partial x} = m_C + O(\Delta x^2) \tag{2.11}$$

$$m_L = \frac{q_i - q_{i-1}}{\Delta x} \qquad \frac{\partial q}{\partial x} = m_L + O(\Delta x) \tag{2.12}$$

$$m_R = \frac{q_{i+1} - q_i}{\Delta x} \qquad \frac{\partial q}{\partial x} = m_R + O(\Delta x). \tag{2.13}$$

Only the *central slope* m_C is of the desired second order accuracy. Unfortunately, strict usage of m_C for the construction of the interface states leads to an unstable scheme. The introduction of new local extrema due to the interpolation as it is shown in the first panel of Figure 2.3 can cause unphysical oscillations with growing amplitudes and thus lead to a code crash. This problem is solved by limiting the slope of u in regions of high curvature. However, this comes at the cost of loosing one order of accuracy in regions where the slope is limited. The aim is thus to deviate from the unlimited central slope m_C as little as possible but still enough to stabilize the numerical scheme. The most defensive slope limiting strategy in RAMSES is the so-called MinMod limiter (see Figure 2.3, second panel).

[6]The deviation from constant values throughout the cells breaks the self-similarity of the Riemann problem.The fluxes at the cell interfaces are therefore not constant in time anymore. The effective input states for the Riemann solver are thus the *predicted* values at the center of the time step, resulting in the fluxes $\mathcal{RP}[U_{i,R}^{n+1/2}, U_{i+1,L}^{n+1/2}]$. R, L denote the right and left boundary states of a cell

The resulting slope is given by

$$m_{\rm MinMod} = \begin{cases} 0 & \text{if } m_L m_R \leq 0, \\ \min(m_L, m_R) & \text{if } m_L, m_R > 0, \\ \max(m_L, m_R) & \text{if } m_L, m_R < 0. \end{cases} \tag{2.14}$$

If both, the left- and right hand slope at a given position have the same sign, the slope with the smaller modulus is chosen. If their signs differ (cell i hosts a local extremum of q), the slope is set to zero. The resulting numerical scheme is second order accurate only in regions where the slope of q is constant, $m_L = m_C = m_R$. A more aggressive strategy is the use of the the MonCen[7] limited slope,

$$m_{\rm MonCen} = \begin{cases} 0 & \text{if } m_L m_R \leq 0, \\ \min(2m_L, m_c, 2m_R) & \text{if } m_L, m_R > 0, \\ \max(2m_L, m_c, 2m_R) & \text{if } m_L, m_R < 0. \end{cases} \tag{2.15}$$

As for the MinMod limiter, the slope is set to zero in cells hosting a local extremum. For all the other cells, the slope is limited such that the resulting interface value is bounded by the average value of q in the neighboring cell (see Figure 2.3, third panel). This results in an unlimited slope and thus second order accuracy in monotonous regions of moderate curvature. The choice of the slope limiter depends on the specific simulation setup. Using the MonCen limiter resulted in a stable simulation in some of the simulations I have performed during the last years. However, cold, highly supersonic flows subjected to strong tidal fields (as they are for example caused by massive sink particles) often required switching to the MinMod limiter.

2.3. Improved Interpolation at Level Boundaries

The slope limiters introduced in the last section are also used to perform interpolation at refinement level boundaries. The fine-level hydrodynamics solver requires a one cell thick layer of "buffer cells" around the refined regions. These cells are only temporarily created and their hydro variables are interpolated from the underlying coarse level. The same interpolation scheme is used whenever a cell is newly refined to define the initial cell-averaged fluid variables in the created cells. Interpolation at level boundaries must therefore be performed in a mass, momentum and energy conserving way from the father cell values. This can be achieved by defining the slope of each conservative variable inside a coarse cell and then using the slopes to construct the values of the conservative variables at the new cell center locations. Again, the central slope needs to be limited *at least* for the density and the energy density in cases where negative values could otherwise result from the interpolation. In practice, it turns out that in many cases not even the use of a MinMod limited slope for the conservative variables is enough to produce a numerically stable scheme. A reason for this numerical unstable behavior is sketched in Figure 2.4. Interpolation of the conservative fluid variables with limited slope does not prevent the introduction of new extrema in the velocity. In cold gas, this can artificially trigger supersonically divergent flows resulting in negative densities. The negative densities then usually trigger a code crash.

[7] The abbreviation MonCen for "monotonized central" does not mean that the reconstructed solution at time t^n is monotonous, but that the resulting scheme is *monotonicity preserving* (see Toro (2009))

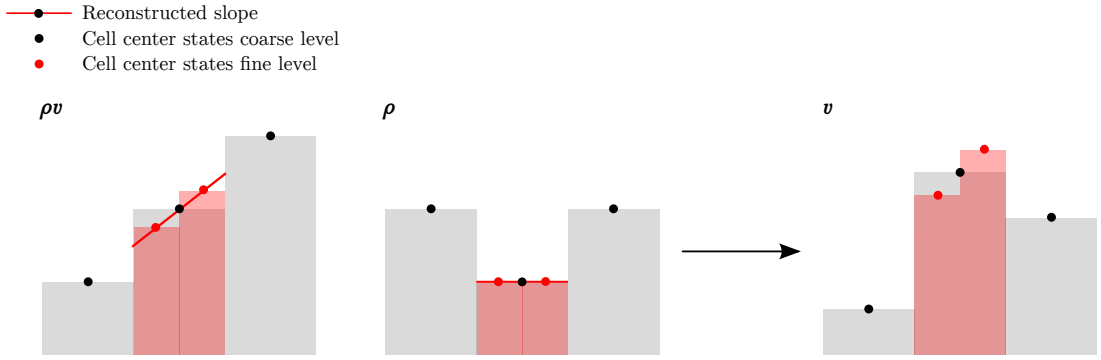


Figure 2.4: Interpolation at level boundaries of density and momentum density. The resulting velocity in the refined cells can have new extrema despite the use of slope limiters on the conservative variables.

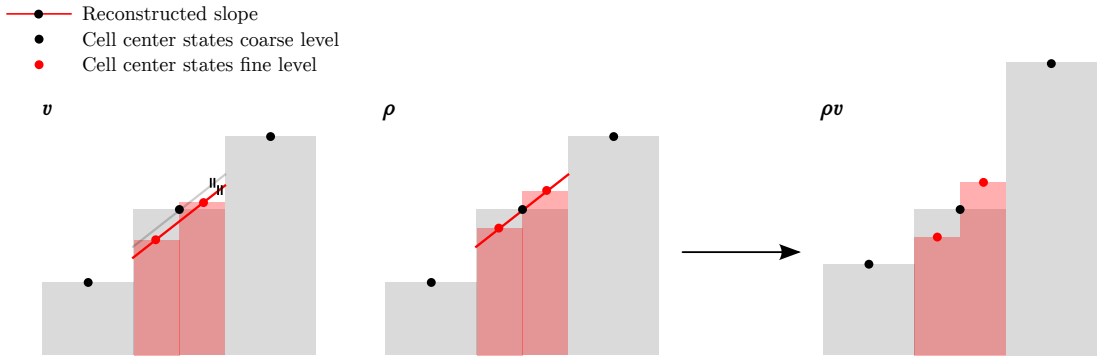


Figure 2.5: Interpolation at level boundaries of density and momentum density. The velocity needs to be corrected in order to conserve momentum.

We have found that interpolating the primitive rather than the conservative fluid variables improves the stability of the scheme. A problem in doing so is that interpolation of the primitive variables does not automatically conserve momentum and energy. We therefore compute the slope of the velocity inside a coarse cell and then adjust the velocities to conserve momentum while keeping the slope fixed. This is sketched in Figure 2.5. Additionally, we perform slope limited interpolation of the thermal energy density which is therefore conserved. Note that the total energy is not conserved as the kinetic energy may change due to the interpolation of density and velocity. In fact, the situation complicates in the case where one requires conservation of the total energy. To achieve this, interpolation could be performed on the total energy density with a limited slope to prevent negative total energy densities. In this case it is the thermal energy density that can become negative as the kinetic energy density may increase due to the interpolation. This problem could be solved by reducing the slope of the velocity in those cases. However, for the simulations performed in this thesis, energy is never strictly conserved anyway as there is always a minimum temperature enforced in some way. We thus ignore the non-conservation of the total energy. In Section 2.5 we test the effect of the chosen interpolation scheme at level boundaries onto the dissipation of momentum in a super- to transsonic turbulent box.

2.4. A new Refinement Criterion for Supersonic Turbulence

We have seen in Section 2.2.1 that in regions of large curvature in the fluid variables, the MUSCL scheme falls back to first order accuracy in space. In the limit of a small time step, the *modified equation* for the momentum density in the first-order 1D Godunov scheme reads (Toro, 2009):

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \frac{D}{2}\Delta x \frac{\partial^2 u}{\partial x^2} \tag{2.16}$$

Figuratively spoken, the modified equation is the partial differential equation that is *actually* solved by the numerical scheme. It arises from the original partial differential equation that one *would like* to solve through an analysis of the local truncation error. The above equation shows that for a first order scheme, truncation errors add an artificial diffusion term to the hydrodynamics equation. This effect is denoted as *numerical diffusion*. The corresponding coefficient is determined by the grid spacing Δx and the maximum wave speed which is determined by the fluid velocity v and the sound speed c_s as

$$D = |v| + c_s. \tag{2.17}$$

A way to decrease the amount of numerical diffusion in the slope-limited regions is to decrease the grid spacing Δx. In an AMR code this can be achieved simply by locally adding a level of refinement. We have thus implemented a curvature-based refinement strategy where we try to minimize diffusion of momentum given the available computational resources. We define the "total momentum diffusion" S_{mom} as the integral of the diffusive term over the entire computational domain,

$$S_{\mathrm{mom}} = \int_{\mathrm{box}} \frac{D}{2}\Delta x \left|\frac{\partial^2 u}{\partial x^2}\right| \mathrm{d}V, \tag{2.18}$$

where u is now specified to be the momentum density. The discrete approximation of the above formula is

$$S_{\mathrm{mom}} \approx \sum_{i \in \mathrm{cells}} \frac{D}{2} \Delta x |u_{i+1} + u_{i-1} - 2u_i| \Delta x^2. \tag{2.19}$$

Now, the goal is to refine those cells which contribute most to the above sum until the allowed "budget" of leaf cells is used. To turn this into an applicable refinement criterion, we specify a desired number of leaf cells N_{goal} in the calculation and an initial guess for a refinement parameter ϵ_{curv}[8] at runtime. We generalize the above considerations to multiple dimensions and refine every cell specified by a tuple of indices $\boldsymbol{i}$ if

$$(|v^n| + c_s)\left|u^m_{\boldsymbol{i}+1_n} + u^m_{\boldsymbol{i}-1_n} - 2u^m_{\boldsymbol{i}}\right| \Delta x^2 > \epsilon_{\mathrm{curv}} \tag{2.20}$$

for any combination of integers n, $m \leq ndim$. In the above formula, $\boldsymbol{i} + 1_n$ denotes the increase of the n-th element of the tuple $\boldsymbol{i}$ by 1. After every coarse time step in the simulation, the refinement parameter ϵ_{curv} is updated based on the new number of leaf cells. If there are too many leaf cells, ϵ_{curv} is increased in order to trigger less refinement, and in case of too few leaf cells, it is decreased. In practice, given a present number of leaf cells N_{leafs}, modifying ϵ_{curv} as in the formula

$$\epsilon^{\mathrm{new}}_{\mathrm{curv}} = \epsilon^{\mathrm{old}}_{\mathrm{curv}} \frac{\log_{10}(N_{\mathrm{leafs}})}{\log_{10}(N_{\mathrm{goal}})} \tag{2.21}$$

has proved to keep the number of leaf cells within $\approx 1\%$ of N_{goal} in the tests we have performed.

2.5. A Testcase for Interpolation- and Refinement Strategy

In this subsection I present a small numerical experiment designed to test the two code modifications described in Sections 2.3 and 2.4. Both modifications aim to reduce numerical diffusion in AMR simulations when shocks and discontinuities are present but not systematically resolved. We thus simulate a periodic box of a supersonically turbulent gas and compare the AMR runs to a simulation where the entire box is resolved at the minimum grid spacing allowed in the AMR runs. We then analyze how the choice of the refinement criterion and the interpolation scheme used at refinement level boundaries influence the dissipation of kinetic energy and the ability to maintain the turbulent cascade to small scales.

2.5.1. Setup

A periodic box filled with isothermal turbulent gas with a velocity power spectrum $P(k) \propto k^{-4}$ and Mach number $\mathcal{M} = 32$ is initialized on a 512^3 grid. The initial density is uniform and no gravitational forces are considered. The simulation is evolved at full 512^3 resolution until the kinetic energy has dropped to 25% of the initial value ($\mathcal{M} \approx 16$) and a log-normal density probability density function has fully developed. This snapshot is used

[8] In practice, a reasonable initial value for ϵ_{curv} is found easiest by picking a random value and running the simulation for a few time steps until ϵ_{curv} has stabilized. This value can then be used as the initial value that should prevent wild over- or under refinements early in the subsequent simulation.

as initial condition for the following experiment.[9] We let the turbulence decay further, thereby comparing the curvature-based refinement criterion (acronym"curv") described in Section 2.4 to a gradient-based method (acronym "grad") and the fixed grid case. In the gradient-based method, a cell is refined whenever the relative variation of a fluid variable between two neighboring cells exceeds a user-defined threshold value. For comparability to the curvature-based criterion, we apply this criterion to the three momentum densities. Furthermore, we adapt the threshold value to trigger the gradient-based refinement in the same way as we do it for the curvature-based criterion (see Equation 2.21). The allowed number of leaf cells for both refinement methods is set to 10 % of the fully refined case, $N_{\rm goal} = 0.1 \times 512^3$. The result of the experiment will therefore tell which method deals better with the limited resources at hand. The second item entering the comparison is the interpolation scheme at level boundaries. With interpolation type "0" we denote straight injection of the coarse values into the refined cells. Interpolation type "2" denotes linear interpolation on the primitive variables as described in Section 2.3 using the MonCen limited slope. As a reference case we run a simulation where the entire grid is refined to the maximum AMR level, denoted as "no AMR". To judge the quality of the different AMR results, we perform a visual comparison of the resulting density structure. Furthermore, we compare the resulting density power spectrum $P_\rho(k)$ and the kinetic energy spectrum $E(k)$ as defined in Kritsuk et al. (2007) to the no AMR case.

2.5.2. Results

We compare the results of the different combinations of refinement criterion and interpolation method at two times. The first snapshot is taken when the sonic scale (computed for the no AMR case) is approximately four times the minimum cell spacing which corresponds to a total Mach number $M \approx 8$. Figure 2.7 shows the density structure of the gas which is dominated by small shocks. While the large scale structure looks very similar for all the cases the runs using a curvature-based refinement strategy contain more of the small-scale features seen in the fully refined reference run. This impression is confirmed by the density power spectrum and kinetic energy spectrum for this snapshot (Figure 2.8 left panels). For both spectra the AMR runs show an excess of power on intermediate scales and deficit in the small scale power. To both directions, the deviation from the no AMR case is a factor of ~ 2 smaller for the curvature-based refinement strategy than for the gradient-based one. The effect of the interpolation method at AMR level boundaries is a little less pronounced. However, especially switching from no interpolation at all to interpolation using the MonCen slope helps to reduce the AMR-introduced deviations from the fixed grid run.

Evolving the simulations further for another ≈ 2 turbulent crossing times lets the Mach number decay to $\mathcal{M} \approx 2$. Fluctuations on scales $\lesssim 64\Delta x_{\rm min}$ are now mostly subsonic. The large scale features seen in Figure 2.7 are still surprisingly similar in all cases. The right panels of Figure 2.8 again show an excess of power at scales around the sonic length

[9] Note that this experiment is not a state-of-the-art simulation of supersonic turbulence. In simulations aiming to reproduce a section of the turbulent inertial range to measure turbulent scaling properties, the turbulence is usually driven for several dynamical times to allow the turbulence to reach a steady state. Furthermore, the resolution of 512^3 is insufficient to capture intermittent regions at the high Mach numbers considered, (see Kritsuk et al. (2007) for more details). However, these simulations are computationally much more expensive than our experiment which has a different purpose. For a comparison of refinement strategies, we consider this setup to be sufficient.

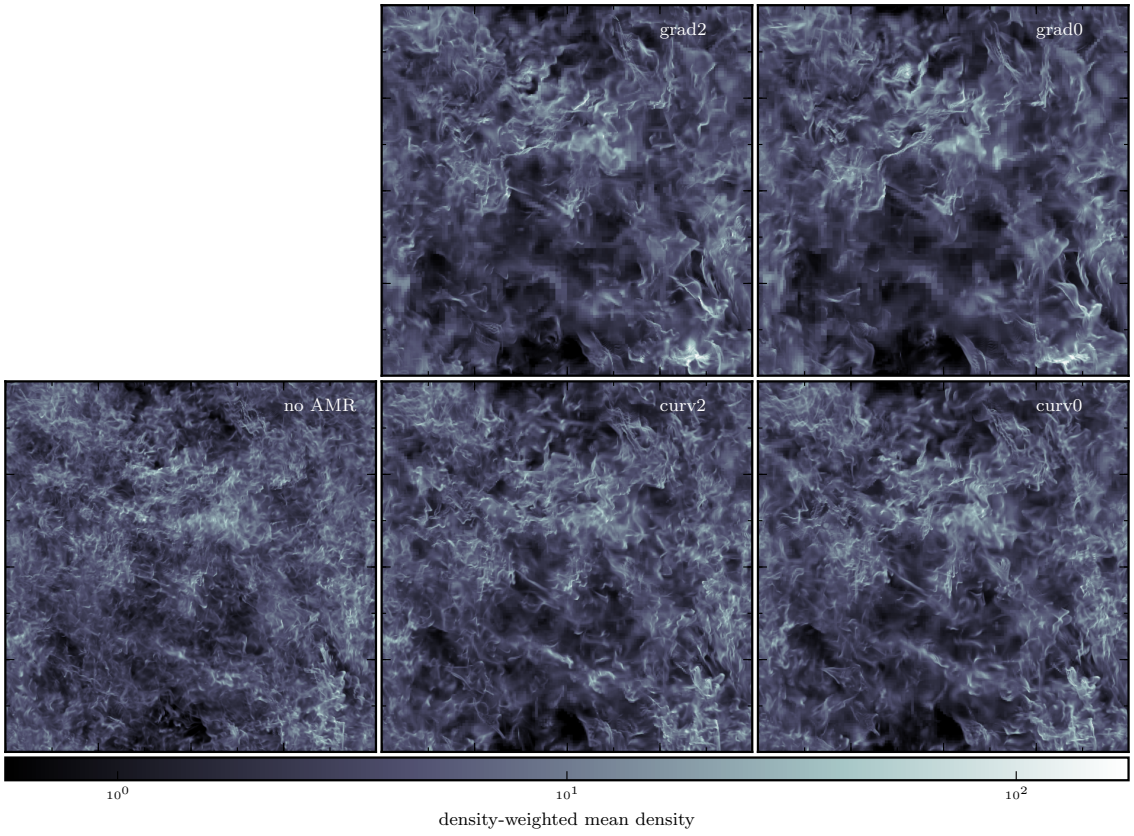


Figure 2.6: Density structure of highly supersonic ($\mathcal{M} \approx 8$) turbulence in the absence of gravity for different refinement and interpolation schemes.

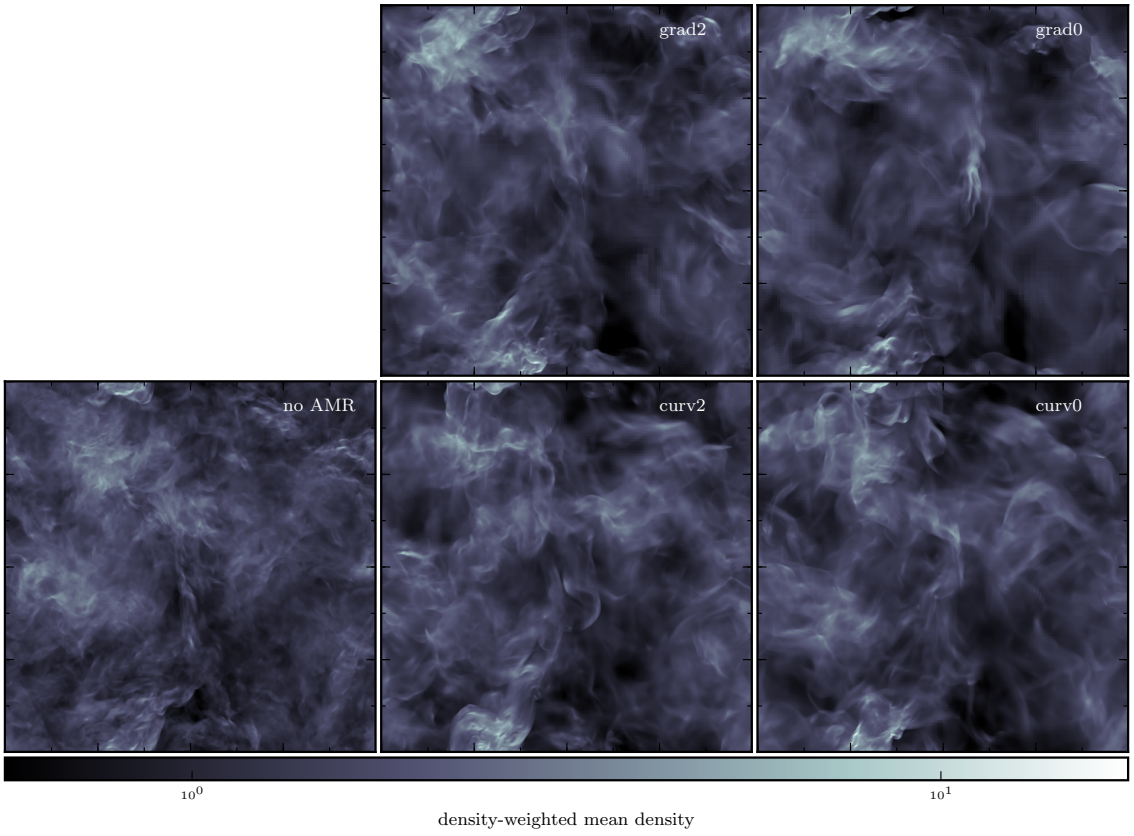


Figure 2.7: Density structure of mildly supersonic ($\mathcal{M} \approx 2$) turbulence in the absence of gravity for different refinement and interpolation schemes.

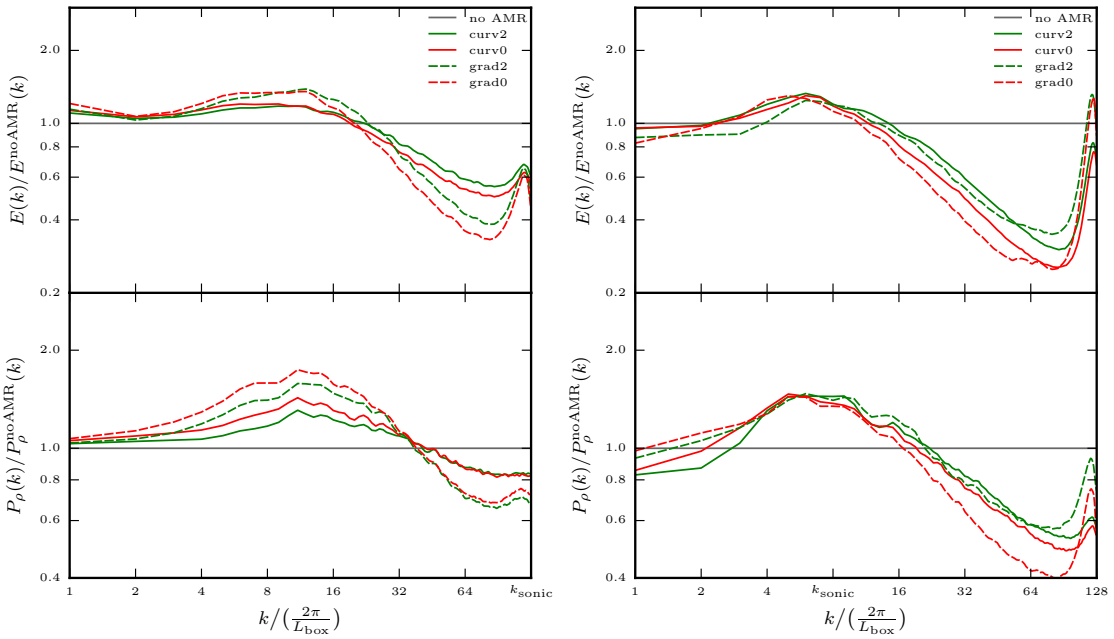


Figure 2.8: Density and kinetic energy power spectra for the two snapshots at $\mathcal{M} \approx 8$ (left Panel) and $\mathcal{M} \approx 2$ (right Panel) for the AMR runs compared to the fully refined reference run. All power spectra are plotted up to wave numbers corresponding to fluctuations on the 128^3 grid which is fully refined in all cases.

and an even bigger lack of power on small scales. The advantage of the curvature-based criterion over the gradient-based one is not as pronounced anymore as it was at higher Mach numbers. In turn, it is now the interpolation at level boundaries which takes a more important role.

2.5.3. Conclusions

In this section I have presented a small test case for the interpolation schemes at AMR level boundaries and the refinement strategy in the case of isothermal, supersonic non-gravitating turbulence. For the highly supersonic case, the new curvature-based criterion performs better in reproducing the statistical properties of the fully refined reference run. Interpolation at level boundaries further decrease the deviation from the reference case. In the subsonic regime, the two refinement criteria perform similarly well, while the advantage that comes from interpolation at level boundaries persists. We therefore use the interpolation scheme as described in Section 2.3 as a default for our star formation simulations. In the majority of the calculations performed so far, using the MonCen slope for interpolation did not result in numerically unstable situations. In cases of heavy sinks embedded in a cold, highly turbulent gas, there have been occasions where switching to the MinMod slope was required. Note that interpolation of the conservative variables was not possible at all for most of these simulations which forced us to use straight injection of the coarse values into the fine level cells.

The situation is different for the new curvature-based refinement criterion. Despite the clear advantage over the gradient-based method in the supersonic regime, we have not

used this criterion for any production simulations so far. As we will show in Section 4.3, once the sound scale and the local Jeans length are resolved, further refinements do not influence the resulting properties of the sink particles formed in a turbulent molecular cloud. Resolving the sound scale is computationally feasible for the molecular clouds up to $1000\,\mathrm{M}_\odot$ that we have simulated in the course of this thesis work. For simulations of much larger molecular clouds where resolving the the sound scale becomes increasingly computationally expensive, the curvature-based refinement criterion might help to reduce computational costs without altering the numbers and properties of the formed molecular cloud cores.

2.6. The Poisson Solver

The standard solver in RAMSES for the Poisson equation is an iterative relaxation method using the multigrid technique (Guillet & Teyssier, 2011; Teyssier, 2002). This is a straightforward choice for an AMR code like RAMSES since it benefits from the AMR hierarchy in natural way. In refined regions, the coarse level potential is interpolated to the fine level cells where it is used as an initial guess and to set boundary conditions. Thanks to the coarse level solution, the initial residual on the fine level contains only high-frequency modes which are known to converge fastest in an relaxation scheme (Press et al., 2007). Furthermore, this approach is well suited for parallelization by decomposing the computational domain as is requires only a one-cell thick boundary layer from the neighboring domains on each AMR level. After solving the Poisson equation, the gravitational accelerations are computed using the five point finite-difference approximation of the gradient Teyssier (2002).

Before the Poisson equation is solved, the source term ρ needs to be constructed. For simulations of self-gravitating gas only, this is trivial as the total density is equal to the gas density. In simulations using particles, the particle density field is computed by projecting the particle density onto the grid using a cloud-in-cell (CIC) or triangular shaped cloud (TSC) scheme Hockney & Eastwood (1981). These schemes transform a discrete particle distribution into a continuous density field by assigning a "shape function" $W(\boldsymbol{x})$ to each particle,

$$\rho_{\mathrm{particle}}(\boldsymbol{x}) = \sum_i m_i W(\boldsymbol{x} - \boldsymbol{x}_i), \tag{2.22}$$

where the index i is running over all particles with masses m_i and positions $\boldsymbol{x}_i$ is the position of particle i. The resulting density field ρ_{particle} is then averaged over the cells to obtain the source term for the Poisson equation. The shape functions for CIC and TSC in one dimension are shown in Figure 2.9. After solving the Poisson equation with the combined gas and particle density as a source term, the gravitational field is interpolated back onto the particle positions using again the same mass assignment scheme. The method of computing inter-particle forces on a grid is known as the particle mesh technique (PM, Hockney & Eastwood (1981)).

2.7. Self-force in the Case of Adaptive Time Stepping: Problem and Proposed Solution

The technique of adaptive time stepping implemented in RAMSES allows finer levels to subcycle coarser ones. This means that the fine level is advanced twice while the coarse level

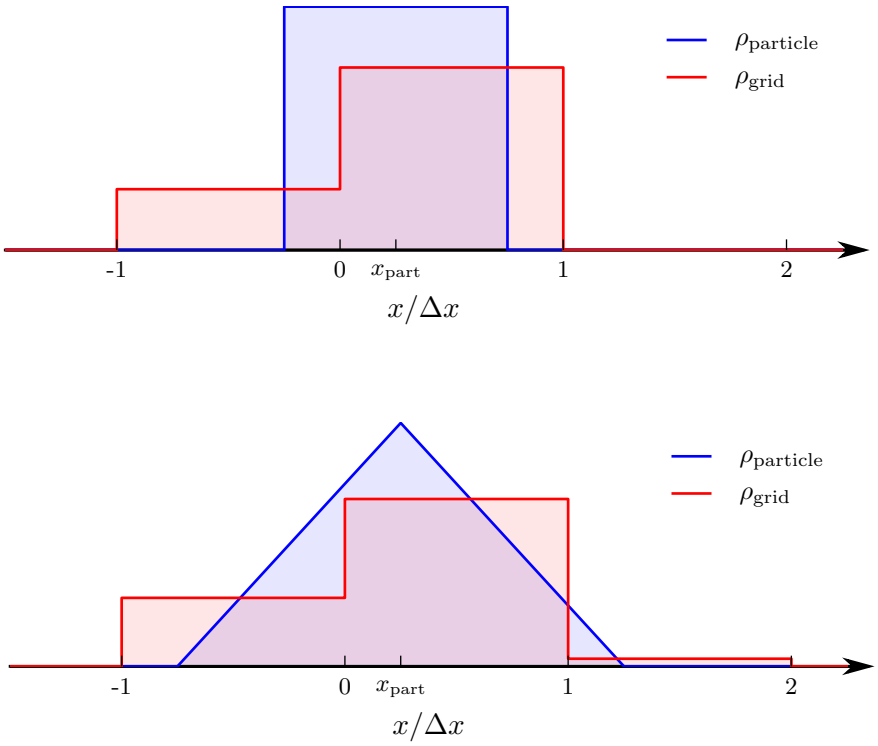


Figure 2.9: Sketch of the CIC (top panel) and TSC (bottom panel) mass assignment schemes in one dimension. The blue lines indicate the respective shape functions of a particle of unit mass sitting at position x_{part}. The resulting discrete density field is shown by the red lines.

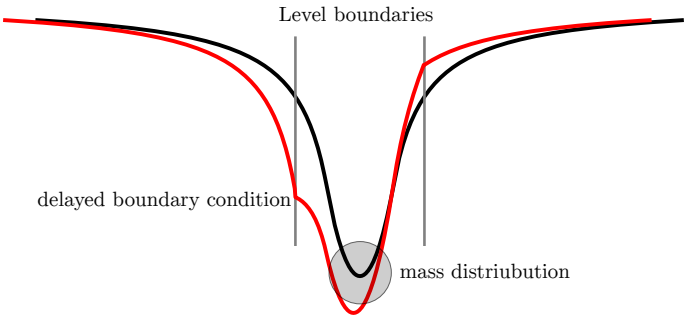


Figure 2.10: Sketch of the self-force felt by a compact massive object as it is moving through an AMR grid in the case of adaptive time stepping. The black line shows the true physical potential caused by the mass distribution. The red line is a combination of the delayed potential on the coarse level and the fine level potential resulting from the asymmetric boundary conditions at the AMR level boundaries.

is advanced only once with a time step equal to the sum of the two small level time steps. This technique can massively speed up simulations with deep AMR hierarchies where only a small fraction of the cells belong to the finest level. A problem of adaptive time stepping lies in the fact that the boundary conditions for the fine-level Poisson equation which are set by the coarser level are "lagging behind" for every other fine-level time step. To illustrate how this can corrupt the result of a simulation we consider a situation where the local gravitational potential is dominated by a highly concentrated object (such as a sink particle) which is moving relative to the grid. Only the high mass region is refined, which causes the refined patch to move along with the sink. In such a situation, the gravitational potential should be symmetric with respect to center of the sink which thus should not feel any gravitational force. However, the "delayed" boundary conditions are asymmetric with respect to the updated sink position which breaks the symmetry of the fine level solution and causes the self-force to act as a spurious drag force. The effect is sketched in Figure 2.10.

This effect which is very small for well resolved, smooth density distributions, but becomes more noticeable when extreme density variations on small scales are present. The prime example of this is the integration of sink particles using the PM method. Sink particles are designed to absorb collapsing gas and can thus grow until they concentrate a considerable fraction of the entire mass present in the simulation in just a few cells. This can lead to the point where self-forces dominate the true gravitational acceleration of the sinks caused by surrounding matter. To alleviate this issue, we have implemented a linear forward extrapolation in time of the coarse level potential to get a better estimate for the fine level boundary conditions. Whenever the coarse level potential is needed at a time where the coarse and the fine level are not synchronized, the coarse potential is created by assuming the same rate of change for the upcoming time step as for the last one. The main computational cost of this approach is the fact that the previous gravitational potential needs to be stored for every cell in the computational domain.

2.8. Self-force tests

We have tested the aforementioned fix for the self-force by comparing the fixed and unfixed method to the respective case where single time stepping was enforced. For this compari-

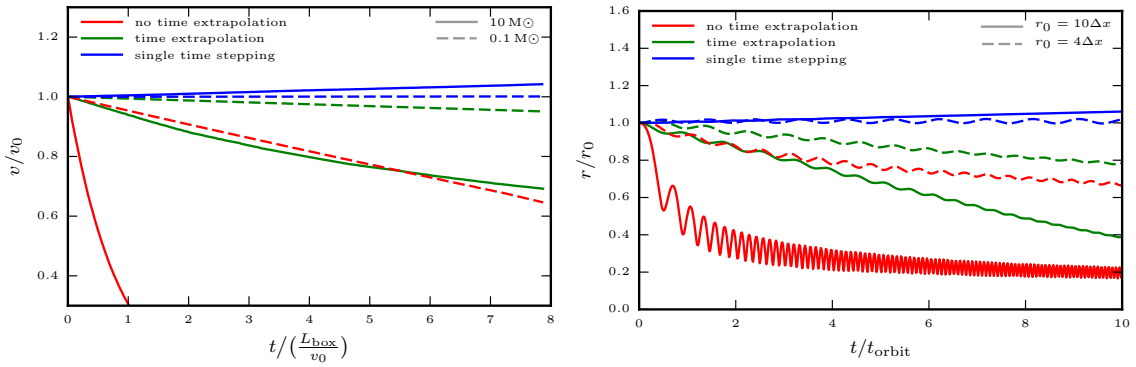


Figure 2.11: Tests of our time extrapolation fix of the self-force. *Left Panel:* An isolated sink is moving through a periodic box and slowed down by the ghost drag. *Right panel:* Two sinks orbiting each other on initially circular orbits. The orbits decay due to the ghost drag force. Both tests show that the time extrapolation fix for the boundary potential does reduce the kinetic energy loss but not entirely fix the problem. The cases where single time stepping is employed show a smaller but nevertheless worrying behavior. Here, boundary effects tend to artificially *add* energy to the system.

son, we have used two simple setups: a single sink particle moving on a straight trajectory and two sinks orbiting each other, both in the absence of gas. In the two cases, we use a 32^3 based grid with one additional level of refinement. The minimum grid spacing is 25 AU[10]. The boundary of the refined region is typically 6 cells away from the center of the sink particle. The initial velocity of the single sink test is $1.5\,\mathrm{km\,s^{-1}}$ and the boundary conditions are set to periodic. We perform the test for two sink masses, $M = 0.1\mathrm{M}_\odot, 10\mathrm{M}_\odot$. The orbit test was performed for two initial separations of the two sink particles. An initial radius of $r_0 = 10\Delta x = 250\,\mathrm{AU}$ corresponds to an initial separation of 20 grid cells which means that the refined regions around the two sinks are not touching. In the case of a binary with smaller initial radius $r_0 = 4\Delta x = 100\,\mathrm{AU}$ the two refined regions around the sinks are connected. The sink masses in the binary test are $1\,\mathrm{M}_\odot$.

The results of both tests are shown in Figure 2.11. Both tests show that the effect of the self-force can be quite dramatic. The isolated sink of $10\,\mathrm{M}_\odot$ sink looses half of its initial velocity within less than one box size (1600 AU) and the wider of the binaries shrinks by 50% within two orbits. The situation is less severe for the less massive sink and the tighter binary where the refined regions around the two sinks are touching each other. The proposed fix to the ghost drag problem reduces the energy loss by roughly one order of magnitude for both test setups. Despite this improvement, orbits of binaries as they are produced in fragmenting dense cores can still not be integrated over the necessary timescales using the PM scheme and adaptive time stepping. A possible way out of this problem is to give up on adaptive time stepping. The price to pay are - depending on the exact setup - much higher computational costs. Additionally, both tests we have performed show that in the case of single time stepping, the presence of level boundaries tends to artificially *inject* energy into the system (see the blue lines in Figure 2.11). This is arguably even more worrying than a small loss rate of kinetic energy. Although we have found the time extrapolation to improve the PM technique for the integration of sink particles when

[10]The scales given here are picked to be representative for a star formation simulation. The results can be rescaled to obtain an estimate for the effect of the self-force in other simulations.

using adaptive time stepping, the numerical experiments presented in this Section clearly show the limitations of the PM approach - a method designed for self-gravitating fluids - in dealing with collisional dynamics.

In Appendix A of our sink method paper (see Chapter 3) we show how computing sink-gas and sink-sink interactions as the direct sum of all pairwise forces solves the issues discussed in this section. As we will show there, the algorithmic scaling of this technique is not satisfactory as there are $n_{\rm sink} \times n_{\rm cells}$ sink-gas interactions to compute. Further improvements could be achieved using fast convolution methods: Passy & Bryan (2014) showed in a recent paper that their application in a PM framework can lead to more satisfying results than the standard PM method for setups like the one we have described in this section. Another interesting variant of fast convolution techniques is the fast multipole method, which could be used to compute sink-gas and sink-sink forces. Fast multipole methods have recently been demonstrated to be fast and accurate also for collisional dynamics (Dehnen, 2014).

3

Paper I: Sink Particle Method Paper

The material presented in this chapter has been published and can be found under the following reference: *Towards a more realistic sink particle algorithm for the RAMSES CODE. Andreas Bleuler; Romain Teyssier. Monthly Notices of the Royal Astronomical Society 2014 445 (2): 4015-4036*

3.1. Abstract

We present a new sink particle algorithm developed for the Adaptive Mesh Refinement code RAMSES. Our main addition is the use of a clump finder to identify density peaks and their associated regions (the peak patches). This allows us to unambiguously define a discrete set of dense molecular cores as potential sites for sink particle formation. Furthermore, we develop a new scheme to decide if the gas in which a sink could potentially form, is indeed gravitationally bound and rapidly collapsing. This is achieved using a general integral form of the virial theorem, where we use the curvature in the gravitational potential to correctly account for the background potential. We detail all the necessary steps to follow the evolution of sink particles in turbulent molecular cloud simulations, such as sink production, their trajectory integration, sink merging and finally the gas accretion rate onto an existing sink. We compare our new recipe for sink formation to other popular implementations. Statistical properties such as the sink mass function, the average sink mass and the sink multiplicity function are used to evaluate the impact that our new scheme has on accurately predicting fundamental quantities such as the stellar initial mass function or the stellar multiplicity function.

3.2. Introduction

Astrophysical simulations of self-gravitating gas often involve regions of gravitational collapse. Resolving those collapses while still following the large scale evolution of the gas therefore requires a huge dynamic range in the density. The local free fall time $t_{\rm ff} = \sqrt{3\pi/32G\rho}$ is a good estimate for the relevant timescales of the dynamics at a given density. For example, a density contrast of 10^{10} observed in giant molecular clouds from the entire cloud down to the first hydrostatic core (Stahler & Palla, 2008) translates into a factor 10^5 between the smallest and the largest timescale of the problem. Advancing the whole simulation at the smallest time step therefore lets the large scale motions appear completely frozen. Adaptive time stepping that allows for different resolution elements to be updated with different time steps (see Bate et al. (1995) for a description in SPH, Teyssier (2002) for AMR) increases the computationally achievable dynamic range in timescales, but long term evolution of systems hosting sites of gravitational collapse is still not possible in many cases. In addition to the problem of time scales, following the collapsing regions to higher densities requires an ever increasing spatial and mass resolution which increases the necessary number of resolution elements in the simulation. It is therefore inevitable to define a maximum resolution at which one does not follow the ongoing collapse any further. Introducing a maximum resolution raises another problem: as Truelove et al. (1997) have shown, not resolving the Jeans length and Jeans mass in regions of gravitational collapse can lead to artificial fragmentation of the gas. A possible way to avoid this is changing the physical model in a way that will artificially stop the gravitational collapse at a scale that can still be resolved. This is usually achieved by implementing a barotropic equation of state that strongly heats the gas once a certain density is exceeded. Federrath et al. (2010) named this approach "Jeans heating'. A problem of this approach is that objects are kept artificially big and therefore more vulnerable to disruption through shocks and tidal stripping. Another way to deal with limited resolution in simulations of gravitational collapse are sink particles. Instead of artificially stopping the collapse at a chosen scale, sink particles approximate the unresolved small-scale evolution by an immediate collapse onto a point mass. A sink interacts with the remaining gas through gravity and accretion only. Once formed, it is disconnected from the hydrodynamic evolution of the system and can not be destroyed anymore.

Despite the radical approximations that come with the introduction of a sink particle, they are widely used in simulations of star formation and sink particle schemes are implemented nowadays in many simulation codes. Given the Lagrangian nature of sink particles they have first been introduced in smoothed particle hydrodynamics (SPH) codes (Gingold & Monaghan, 1977; Lucy, 1977). It was Bate et al. (1995) who presented the first implementation which most subsequent implementations in SPH codes are based upon, like in the codes GADGET (Jappsen et al., 2005), GASOLINE, (Shen & Wadsley, 2006), DRAGON, (Goodwin et al., 2004) and SEREN (Hubber et al., 2011)). More recently, Hubber et al. (2013) have introduced a more advanced algorithm that deviates quite strongly from the original one by Bate et al. (1995).

Krumholz et al. (2004) were the first to introduce sink particles in the Eulerian, grid-based code ORION, built upon the adaptive mesh refinement (AMR) technique (Berger & Colella, 1989a,b). Their implementation has been the role model for sink particle implementations into various other grid-based codes, such as ENZO (Wang et al., 2010), RAMSES (Dubois et al., 2010), PENCIL (Padoan & Nordlund, 2011) and ORION2 (Lee et al., 2014). Later Federrath et al. (2010) presented their sink implementation into the FLASH code

which deviates considerably from the original Krumholz et al. (2004) method. A more recent implementation has been presented by Gong & Ostriker (2013) for the ATHENA code, quite close to the Federrath et al. (2010) method. While sinks have been used in different context, such as formation and growth of black holes, most of the implementations mentioned above are targeting star formation as the primary application for sink particles.

Simulations of star formation have made tremendous progress throughout the last decade. The increase in computational power and the ongoing evolution of algorithms has allowed simulations of larger volumes and finer resolution. Beyond that, the implementation of radiative transfer, magnetic fields, outflows and chemical evolution models has led to a much better understanding of star formation (e.g., Bate, 2012; Krumholz et al., 2012; Offner et al., 2009; Wang et al., 2010). Some of this additional physics is tightly coupled to the sink particles as they act as a source for feedback processes. This increases the impact of sink particles on the remaining gas. Furthermore, various sink properties such as their mass function, accretion rates, multiplicity fractions and formation rates are used directly for comparison with observations. It is therefore crucial to have reliable sink particle algorithms as well as a good understanding of how the details in the implementation affect the results.

This is precisely the goal of the present paper: we describe a new, possibly better sink particle implementation together with a suite of test cases that we use for comparing the components of our new algorithm to already existing implementations, mostly in AMR codes. The main novelties in our code are related to the formation of sink particles. We run a clump finder to identify well defined density peaks in the gas which are then treated as possible locations for sink formation. We introduce more exact criteria to check whether the gas inside a small volume around such a peak is undergoing gravitational collapse and therefore allowed for sink formation. The paper is structured as follows: in Section 3.3 we present our algorithm for sink formation and discuss differences and similarities to existing codes. In section 3.4 we briefly discuss the issue of sink merging. Section 3.5 deals with numerical methods for the integration of the sink particle trajectories. In Section 3.6 we describe different methods for modeling the accretion of gas onto the sink particles. Finally, Section 3.7 describes the test and comparison cases that we used to test sink formation, sink merging and the accretion onto sink particles. The Appendix contains a comparison of two integration schemes for the sink particles.

3.3. Sink particle creation

The existing implementations of sink particles into AMR codes can be divided into two classes, namely “cell-based” and “peak-based” techniques. In cell-based methods, sink particles are formed based on purely local quantities. By local, we mean gas properties associated to the corresponding cell only. For example, Krumholz et al. (2004) form sinks in every cell with convergent velocity field whose density exceeds a given threshold. This often results in a connected region where every cell forms a sink particle. These sink particles are then merged using a friends-of-friends (FOF) algorithm (Davis et al., 1985). In contrast, peak-based techniques define small volumes around density peaks above a given density, and apply criteria for sink formation based on quantities integrated over such a volume. This “control volume” around a density peak is usually a sphere with radius chosen equal to the accretion radius (Federrath et al., 2010).

Sink particles inevitably introduce a level of discretisation in our continuous fluid description. Cell and peak-based methods can be seen as different approaches to perform

this discretisation. Cell-based approaches form sinks in a continuous, or cell-by-cell way. The discretisation is introduced later by the FOF algorithm, that will break up connected regions into multiple FOF groups. The resulting distribution of the sinks therefore critically depends on the adopted linking length. Peak-based methods introduce discretisation in our fluid by considering only density peaks for sink formation. Note that accretion can affect the results of that procedure by creating a "hole" around the sink and thus creating new artificial sink formation sites close to the boundary of the accretion zone.

Our new method that we label as "clump-based" is an extension of the peak-based method. Instead of considering every density peak for sink formation, including possibly small fluctuations, we require the peak to have a certain *prominence*[1]. Peaks that fail this criterion are considered as "noise" and are merged to neighbouring ones. This provides a more robust segmentation of the volume into a discrete set of subregions, excluding small density fluctuations from the analysis. We consider this as being particularly important if sinks are not allowed to merge during the course of the simulation (see Section 3.4 for more details on sink merging). As in the peak-based approach, we define spherical regions around the candidate locations for sink formation. Those regions are then examined for conditions of gravitational collapse.

This raises the question about the size of the region that should be considered. At first sight, taking the accretion zone (i.e. a sphere of radius $R_{\rm acc} \approx 4\Delta x_{\rm min}$) as the integration domain for further energetic considerations appears as a natural choice, as it contains the gas from which the sink will form. Considering a larger volume might detect gravitational collapse which can still be well resolved by the simulation and therefore should not trigger sink formation yet. Using a smaller volume leads to a poor definition of quantities such as the internal kinetic energy of the gas inside the sphere. In terms of recent theoretical developments on the origin of the IMF (Hennebelle & Chabrier, 2008; Hopkins, 2012), one can say that the sink particle is introduced when the smallest gravitationally bound scale ("last crossing scale") is of the order of the accretion radius. If we pick the sink formation threshold $\rho_{\rm sink}$ in agreement with the Truelove et al. (1997) criterion such that the minimum Jeans mass is resolved by 4 cells at the maximum level of refinement, gravity should start to dominate pressure at the scale of the accretion radius which again justifies the use of a sphere of that size to evaluate gravitational collapse. As we have just mentioned, the minimum grid spacing sets the maximum density in the simulation (or vice-versa). The remaining free parameter can be set by computational or physical arguments. One can simply choose a certain resolution with respect to the computational resources at hand, knowing that one will miss fragmentation into objects smaller than that scale. Another option is to look for a physical scale (such as the opacity limit in molecular gas at $\sim 10^{-13}{\rm g/cm}^3$) to set a minimum scale for fragmentation.

We will now turn to the more detailed description of our new method for sink formation. It consists of the following steps which are described in the following subsections: We check for the creation of new sink particles after every coarse time step.[2] First, we run the clump finder to identify peaks and their associated regions. The peak locations identified by the clump finder are taken into account as possible locations $\boldsymbol{x}_i$ for sink formation. For each of these locations we define a region Ω_i containing all the cells that lie within the accretion

[1] The criterion that we apply is closely linked to the definition of the term *prominence* in topography, see Section 3.3.1 for more details on the clump finder

[2] RAMSES allows adaptive time stepping for cells at different levels. This is achieved by updating a fine cell twice while a coarse cell is updated once with a time step which equals the sum of the two fine-level time steps. After every coarse time step, all the levels are synchronized.

radius from the location considered. The gas inside Ω_i must be undergoing contraction along *all* directions in order trigger sink formation (collapse check). Furthermore, the gravitational field must be strong enough to overcome all internal support in the gas (virial check). If a peak lies within the accretion radius from a pre-existing sink particle, it is not allowed to form a sink (proximity check).

3.3.1. The RAMSES clump finder

Observers have been identifying bound structure in molecular clouds for a long time[3]. Williams et al. (1994) describe an algorithm called CLUMPFIND which finds clumps in a PPV (position-position-velocity) cube using a set of isodensity contours. In this method, a gas clump is identified as such if its highest saddle point[4] is separated from the peak by a contour surface. When operating in log-space with equally spaced contour levels, the contour levels differ by a constant factor in linear space. A clump with a peak-to-saddle ratio above that factor will therefore always be recognized as an individual clump. However, peaks with a lower peak-to-saddle ratio can be separated from their highest saddle point if a contour level happens to be in between the peak and the highest saddle point. Our RAMSES clump finder defines clumps in a very similar way as the method by Williams et al. (1994). The main difference is that we remove the probabilistic element that comes with the introduction of a finite set of contour levels. Instead of contouring the dataset we identify all peaks and their highest saddle points above a given threshold. We then require the peak-to-saddle ratio to be above a certain value for the peak to survive. Otherwise it is merged to the neighbor it shares the highest saddle point with. We now describe our clump finder in more detail. It works by performing the following steps which are sketched in Figures 3.1 - 3.6:

i) In a first step, every cell whose density is higher than a given threshold is marked (Figure 3.2).

ii) Every marked cell is then assigned to a density peak by following the path of steepest ascent. We do this by first checking for every marked cell whether it is a local density maximum.[5] The found maxima are labeled with a global peak-id. All cells above the threshold are sorted in descending density. Next, a loop over all cells is performed where every cell is assigned the peak-id of its densest neighbor. The previous sorting guarantees that the densest neighbor does already have a peak-id assigned. All cells sharing the same peak-id form a so called "peak patch" (Figure 3.3).

iii) The saddle point densities connecting between all peak patches are identified. For this purpose we introduce a sparse, symmetric connectivity matrix M of virtual size $n_{\rm peak}^2$. The value $M(i,j)$ contains the maximum saddle point density connecting peak i with peak j. In order to construct this matrix we check for each cell belonging to a certain clump whether it has a neighbor which belongs to a different clump. If this is the case, the average density of the cell and its neighbor is considered the density

[3]Finding dark matter haloes in cosmological simulations has been also developed for many decades, and is very similar to finding clumps in turbulent gas. Techniques used in halo finders have influenced our clump finder and can be found in various codes such as SUBFIND (Springel et al., 2001) or ADAPTAHOP (Aubert et al., 2004).

[4]In topography this would be called a key col or key saddle.

[5]Note that we consider every cell with a common face, edge or corner as a neighbor of a given cell.

at the common surface and written into the $M(i, j)$ if it is bigger than the existing value. The highest saddle point lying on the boundary of a certain peak patch is the relevant one for our analysis. This corresponds to the maximum of a certain line in the connectivity matrix. By looking at the ratio of the peak density to the maximum saddle density of a peak we decide whether this is a significant one or not. We usually require this peak-to-saddle ratio to be bigger than 2.[6]

iv) The peak patches are sorted by ascending peak density. Insignificant peak patches are merged to the one they are connected to through the highest saddle point. The sorting is important since it makes sure that no peak patch is merged with one that has already been merged into another one before. Isolated peak patches which are insignificant are rejected (Figures 3.4,3.5). After every single merger, we update the connectivity matrix and the peak-to-saddle ratio of the peak patch that has grown due to the merger.

v) After the previous step all insignificant peak patches have been rejected or merged to form significant ones which we now label as clumps (Figure 3.6). The list of mergers is used to link every peak patch initially present (Figure 3.3) to the final clump in the merging history and all cells above the density threshold are reassigned their new peak-id.

Since we want to use our clump finder to find possible locations for sink particle creation, it needs to run on the fly. It is therefore implemented in a parallel fashion. The steps (i-v) need to be adapted in order to the be implemented in a MPI code where every MPI domain only contains a fraction of the whole computational domain. In RAMSES (Teyssier, 2002) the cells that belong to a MPI processes domain ("active" cells) are wrapped in a thin layer of cells that do belong to neighboring MPI domains ('virtual boundaries'). In step (i) only active cells are flagged. In step (ii) the flagged cells are sorted inside each MPI domain and the loop over all cells is performed by each MPI process individually. After this loop, the peak-id of active cells close to a domain boundary are copied into the virtual boundary regions of the neighboring MPI domains and the loop is repeated until every cell is either a local maximum or has the same peak-id as its densest neighbor. In step (iii) we keep the connectivity matrix $M(i, j)$ local to each domain, while the other quantities of the peaks (peak density, peak position) are global in the sense that all MPI processes have the information about all peaks. When clump i needs to be merged, every MPI process searches for its own maximum in the i-th line of $M(i, j)$. The values of all the maxima are compared between the MPI processes to find the index of the global maximum. The mergers in step (iv) and the final link from initial peak-id to final peak-id in step (v) are performed globally by all MPI processes and the actual reassignment of cells with their final peak-id is done by each MPI process for its active cells.

3.3.2. Virial check

The gas surrounding the density peaks found by the clump finder is investigated for gravitational collapse. We perform a virial theorem type analysis to balance the gas configurations self gravity against the gas internal support. As it is done in textbooks when deriving

[6] The exact choice of this value is not critical for the formation of sink particles. The checks which are applied later (see Section 3.3) usually ensure a higher peak-to-saddle ratio than what we require here.

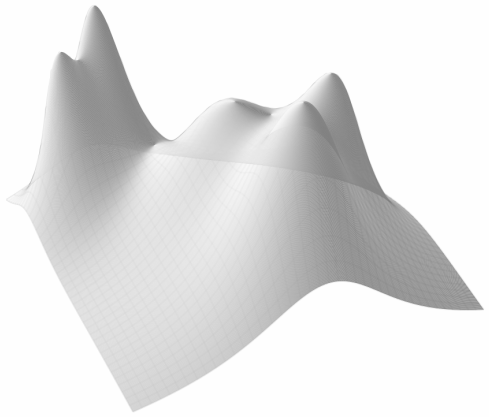


Figure 3.1

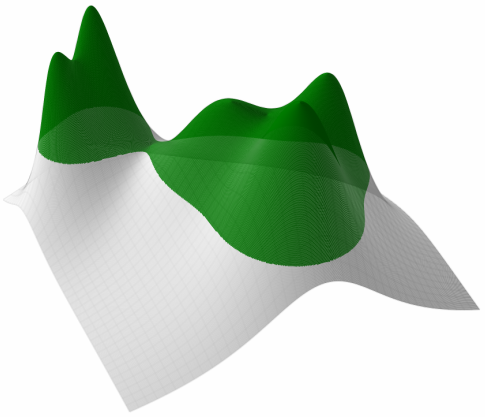


Figure 3.2

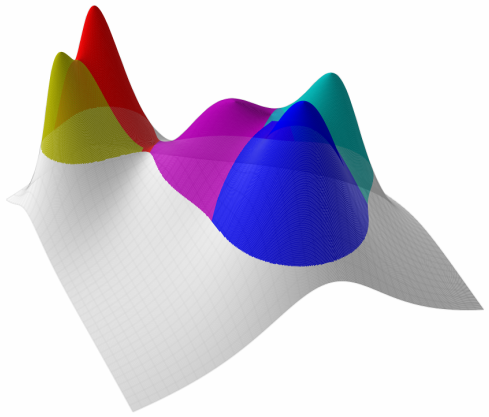


Figure 3.3

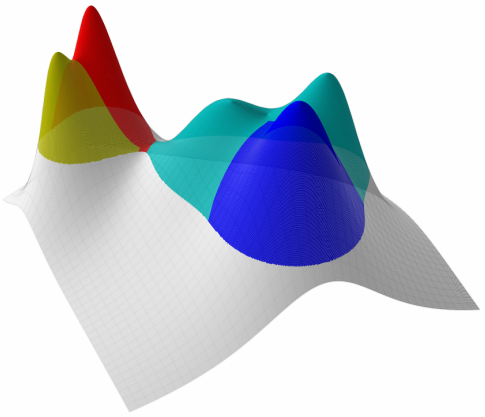


Figure 3.4

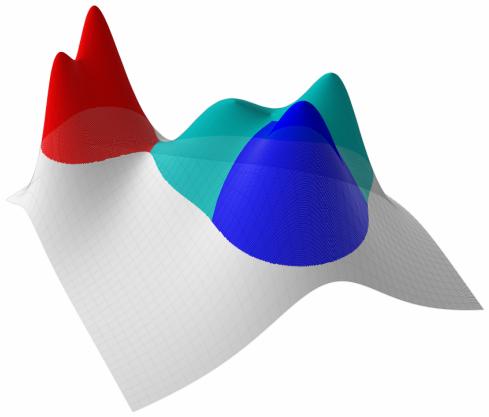


Figure 3.5

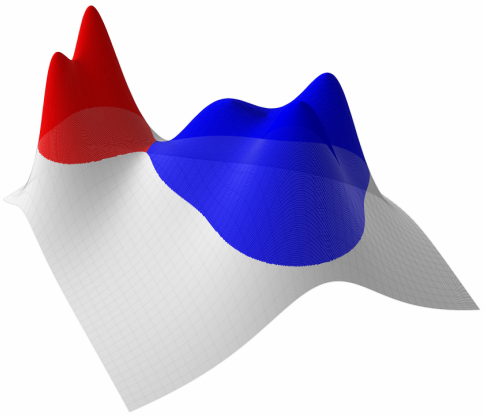


Figure 3.6

Figure 3.1 - 3.6 Working principle of the clump finder represented on a 2d-surface.

the virial theorem (e.g., Stahler & Palla, 2008) we start by defining the scalar moment of inertia

$$\mathcal{I} = \int_{\Omega_i} \rho |\boldsymbol{r}|^2 \mathrm{d}V \tag{3.1}$$

as a measure of the spatial extent of the gas configuration contained in Ω_i. The corresponding acceleration is found computing the second derivative in time of $\mathcal{I}$. Since the volume Ω_i is moving with the flow, we apply Reynolds transport theorem twice to obtain

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{I} = \int_{\Omega_i} \rho |\boldsymbol{v}|^2 \mathrm{d}V + \int_{\Omega_i} \rho \left(\boldsymbol{r} \cdot \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} \right) \mathrm{d}V, \tag{3.2}$$

where the $\mathrm{D}/\mathrm{D}t$ operator stands for the Lagrangian derivative. We now write the Euler equation in Lagrangian form, using gravitational and radiative acceleration as external forces and the general form of the stress tensor $\overline{\overline{\sigma}}$ for internal forces,

$$\rho \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} = \rho \boldsymbol{g} + \frac{\kappa \rho}{c} \boldsymbol{F}_{\mathrm{rad}} + \nabla \cdot \overline{\overline{\sigma}}. \tag{3.3}$$

In the previous equation $\boldsymbol{g}$ stands for the gravitational acceleration and $\boldsymbol{F}_{\mathrm{rad}}$, κ, c are the radiation flux the opacity and the speed of light. Injecting the Euler equation in Equation 3.2 gives

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{I} = \int_{\Omega_i} \rho |\boldsymbol{v}|^2 \mathrm{d}V + \int_{\Omega_i} \rho\, \boldsymbol{g} \cdot \boldsymbol{r}\, \mathrm{d}V \tag{3.4}$$

$$+ \int_{\Omega_i} \frac{\kappa \rho}{c} \mathrm{r} \cdot \boldsymbol{F}_{\mathrm{rad}} \mathrm{d}V + \int_{\Omega_i} \mathrm{r} \cdot (\nabla \cdot \overline{\overline{\sigma}})\, \mathrm{d}V. \tag{3.5}$$

We use the vector identity

$$\nabla \cdot (\overline{\overline{\sigma}} \boldsymbol{r}) = \boldsymbol{r} \cdot (\nabla \cdot \overline{\overline{\sigma}}) + \mathrm{Tr}(\overline{\overline{\sigma}}) \tag{3.6}$$

to obtain the virial theorem in its generalized form,

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{I} = \int_{\Omega_i} \rho |\boldsymbol{v}|^2 \mathrm{d}V + \int_{\Omega_i} \rho\, \boldsymbol{g} \cdot \boldsymbol{r}\, \mathrm{d}V \tag{3.7}$$

$$+ \int_{\Omega_i} \frac{\kappa \rho}{c} \mathrm{r} \cdot \boldsymbol{F}_{\mathrm{rad}} \mathrm{d}V - \int_{\Omega_i} \mathrm{Tr}(\overline{\overline{\sigma}})\, \mathrm{d}V + \int_{\partial \Omega_i} \boldsymbol{r} \cdot (\overline{\overline{\sigma}} \boldsymbol{n})\, \mathrm{d}A. \tag{3.8}$$

We have used the divergence theorem to transform the volume integral over the left-hand term in Equation 3.6 into a surface integral, $\partial \Omega_i$ therefore denotes the boundary of Ω_i and $\boldsymbol{n}$ is the outward pointing unit normal to the boundary. The stress tensor can be written in general for a viscous magnetized fluid as

$$\overline{\overline{\sigma}} = -P \mathbb{1} + \overline{\overline{\tau}} + \overline{\overline{\mathcal{M}}} \tag{3.9}$$

where $\overline{\overline{\tau}}$ is the viscous stress tensor and the magnetic stress is given in the ideal MHD limit by the Maxwell tensor

$$\mathcal{M}_{ij} = B_i B_j - \frac{B^2}{2}\delta_{ij}. \tag{3.10}$$

In the case of isotropic stresses and without radiation, this simplifies into

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{I} = \underbrace{\int_{\Omega_i} \rho|\boldsymbol{v}|^2 \mathrm{d}V}_{\text{kinetic energy term}} + \underbrace{\int_{\Omega_i} \rho\, \boldsymbol{g}\cdot\boldsymbol{r}\, \mathrm{d}V}_{\text{tidal energy term}} \tag{3.11}$$

$$+ \underbrace{3\int_{\Omega_i} P \mathrm{d}V}_{\text{volume pressure term}} - \underbrace{\int_{\partial\Omega_i} P\, \boldsymbol{n}\cdot\boldsymbol{r}\, \mathrm{d}A}_{\text{surface pressure term}}, \tag{3.12}$$

While the first term on the right-hand side is indeed twice the kinetic energy, one must make further assumptions if one wishes to simplify this into a more common form of the virial theorem. The second term is usually identified as the total gravitational energy

$$E_{\text{pot}} = \frac{1}{2}\int_{\Omega_i} \rho\, \phi_g \mathrm{d}V, \tag{3.13}$$

This is valid only if the potential is caused entirely by the gas inside Ω_i. A physically more correct interpretation of this term is obtained using a first order Taylor expansion of the gravity acceleration with respect to the center of mass as

$$\boldsymbol{g} \simeq \boldsymbol{g}_{cm} + \overline{\overline{\mathcal{T}}}\,(\boldsymbol{r} - \boldsymbol{r}_{cm}) \tag{3.14}$$

where $\overline{\overline{\mathcal{T}}}$ is the *tidal tensor*, so that the tidal energy term can be written to leading order as

$$\int_{\Omega_i} \rho\, \boldsymbol{g}\cdot\boldsymbol{r}\, \mathrm{d}V \simeq \boldsymbol{g}_{cm}\cdot\boldsymbol{r}_{cm} + \int_{\Omega_i} \rho\, \boldsymbol{r}_{rel}\cdot\overline{\overline{\mathcal{T}}}\boldsymbol{r}_{rel}\, \mathrm{d}V \tag{3.15}$$

which demonstrates that this term is related to the tidal tensor, not to the potential energy. The third term is equal to the thermal energy only for certain equations of state. Furthermore, the pressure surface term is often neglected. These various approximations might be justified when considering a gas configuration which is (nearly) in isolation. However, the gas from which sinks form is typically far from being isolated and we therefore do not simplify Equation 3.11 any further.

Just as the inertia tensor that arises when studying the dynamics of rigid bodies, the scalar moment of inertia defined in Equation 3.1 depends on the choice of the coordinate system. Starting from the scalar moment of inertia in the center of mass frame, $\mathcal{I}_{\text{cm}}$, we find $\mathcal{I}$, the scalar moment of inertia of the same object with center of mass located at position $\boldsymbol{x}_{\text{cm}}$, using the equivalent of the parallel-axis theorem

$$\mathcal{I} = \mathcal{I}_{\text{cm}} + M|\boldsymbol{r}_{\text{cm}}|^2. \tag{3.16}$$

While the above derivations of the generalized virial theorem hold for any inertial frame of reference, the interpretation of $\mathcal{I}$ as a measure of the size of the gas configuration only

makes sense as long at the coordinate origin is located in the center of mass. We therefore choose the frame which is comoving with the center of mass of the gas contained in Ω_i. This non-inertial frame gives rise to a fictitious acceleration that enters Equation 3.3. As long as the frame is non-rotating, this additional acceleration term is independent of the position in space and its contribution to the second term on the right-hand side of Equation 3.7 vanishes in the center of mass frame[7]. We thus rewrite Equation 3.7 in the center of mass frame

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{I}_{\mathrm{cm}} = \int_{\Omega_i} \rho|\boldsymbol{v}_{\mathrm{rel}}|^2 \mathrm{d}V + \int_{\Omega_i} \rho\, \hat{\boldsymbol{g}}_{\mathrm{rel}} \cdot \boldsymbol{r}_{\mathrm{rel}}\, \mathrm{d}V \tag{3.18}$$

$$- \int_{\Omega_i} \mathrm{Tr}(\overline{\overline{\sigma}})\, \mathrm{d}V + \int_{\partial\Omega_i} \boldsymbol{r}_{\mathrm{rel}} \cdot (\overline{\overline{\sigma}}\boldsymbol{n})\, \mathrm{d}A. \tag{3.19}$$

where the index "rel" refers to the position, velocity and acceleration relative to their centre of mass values. For simplicity we have absorbed the radiation force as an effective gravitational acceleration

$$\hat{\boldsymbol{g}} = \boldsymbol{g} + \frac{\kappa}{c}\boldsymbol{F}_{\mathrm{rad}}. \tag{3.20}$$

It is this last version of the virial theorem that we use as check for sink formation. For an inviscid gas in the absence of radiation and magnetic fields as it is the case in the tests described in Section 3.7, $\hat{\boldsymbol{g}}_{\mathrm{rel}}$ is therefore simply the relative gravitational acceleration $\boldsymbol{g}_{\mathrm{rel}}$ and the stress is given by $\overline{\overline{\sigma}} = -P\mathbb{1}$. Equation 3.18 simplifies to

$$\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathcal{I}_{\mathrm{cm}} = \int_{\Omega_i} \rho|\boldsymbol{v}_{\mathrm{rel}}|^2 \mathrm{d}V + \int_{\Omega_i} \rho\, \boldsymbol{g}_{\mathrm{rel}} \cdot \boldsymbol{r}_{\mathrm{rel}}\, \mathrm{d}V \tag{3.21}$$

$$+ 3\int_{\Omega_i} P \mathrm{d}V - \int_{\partial\Omega_i} P\, \boldsymbol{n} \cdot \boldsymbol{r}_{\mathrm{rel}}\, \mathrm{d}A, \tag{3.22}$$

which is the same as Equation 3.11, but this time in the comoving, non-inertial center of mass frame. Note that the last term in the above equation simplifies to $4\pi R^3 P_{\mathrm{surface}}$ for a spherical region of radius R, which cancels with the volume pressure term in the case of constant pressure. The gas in Ω_i is only further considered for sink formation, if $\ddot{\mathcal{I}}_{\mathrm{cm}} < 0$. This condition ensures that the gravitational field at a possible location for sink formation is compressive and strong enough to overcome all internal support present in the gas. In contrast to estimations of the gravitational potential energy that do neglect the curvature of the background potential, our version fully takes into account any tidal forces that could prevent the collapse of the gas. All the required quantities are readily available in the computational code, which makes this condition well suited for implementation in simulations.

[7] For a spatially constant fictitious acceleration $\boldsymbol{g}_{\mathrm{fict}}$ we have

$$\int_{\Omega_i} \rho\, \boldsymbol{g}_{\mathrm{fict}} \cdot \boldsymbol{r}\, \mathrm{d}V = \boldsymbol{g}_{\mathrm{fict}} \cdot \int_{\Omega_i} \rho\, \boldsymbol{r}\, \mathrm{d}V = \boldsymbol{g}_{\mathrm{fict}} \cdot \boldsymbol{r}_{\mathrm{cm}} \tag{3.17}$$

which vanishes since $\boldsymbol{r}_{\mathrm{cm}} = 0$ in the center of mass frame.

3.3.3. Collapse check

The gas which is about to form a sink particle must not only be accelerated towards the center of the volume under consideration, it must as well be contracting at the moment of formation. Krumholz et al. (2004) require $\nabla \cdot \boldsymbol{v} < 0$ for a cell which is about to form a sink. Federrath et al. (2010) apply a similar check by requiring that the gas inside the "control volume" is contracting along all principal axes. We adapt this criterion to our analysis presented in Section 3.3.2 and compute all eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and the corresponding normalized eigenvetors $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ of the symmetric tensor [8]

$$\overline{\overline{I}}_{\rm cm} = \int_{\Omega_i} \rho \, \boldsymbol{r}_{\rm rel} \otimes \boldsymbol{r}_{\rm rel} \, {\rm d}V. \tag{3.24}$$

By computing the time derivative

$$\frac{{\rm d}\overline{\overline{I}}_{\rm cm}}{{\rm d}t} = \int_{\Omega_i} \rho (\boldsymbol{r}_{\rm rel} \otimes \boldsymbol{v}_{\rm rel} + \boldsymbol{v}_{\rm rel} \otimes \boldsymbol{r}_{\rm rel}) {\rm d}V \tag{3.25}$$

we can assign a collapse timescale to each direction given by the eigenvectors of $\overline{\overline{I}}_{\rm cm}$

$$t_i = \frac{\lambda_i}{(\frac{{\rm d}\overline{\overline{I}}_{\rm cm}}{{\rm d}t} \boldsymbol{e}_i) \cdot \boldsymbol{e}_i}, \tag{3.26}$$

where a small negative timescale indicates fast collapse along a certain axis. Only one negative timescale is a sign for sheet-like and two negative timescales indicate filamentary collapse. Although collapsing, these collapsed regions are poorly approximated by a point mass. We therefore require all three timescales to be negative to ensure collapse onto a point-like object before we introduce a sink particle. This condition can be further strengthened by enforcing collapse along all axis within a certain time (see 3.4.1).

3.3.4. Proximity check

Gas which is falling onto an existing sink particle is not allowed to form another sink, even if there is a density peak which fulfills all criteria for sink formation. We therefore check whether the possible location is closer than one accretion radius from an existing sink. If it is, we do not allow formation of a new sink. Federrath et al. (2010) applied this test that can be seen as the possibility for sinks to merge to existing ones at their time of birth (see Section 3.4).

3.3.5. Alternative checks

We briefly present and discuss alternative checks which we implemented for testing and comparison reasons, but are not used in our final version of the code. All these tests have been described by Federrath et al. (2010) to whom we refer for more details.

[8] Note that

$$(\overline{\overline{I}}_{\rm cm} \boldsymbol{u}) \cdot \boldsymbol{u} = \int_{\Omega_i} \rho \, (\boldsymbol{u} \cdot \boldsymbol{r}_{\rm rel})^2 \, {\rm d}V. \tag{3.23}$$

is a measure of the extension of an object along a certain direction specified by the unit vector $\boldsymbol{u}$.

3.3.5.1. Bound state check

The total energy in the control volume must be negative to form a sink,

$$E_{\rm pot} + E_{\rm kin} + E_{\rm therm} + E_{\rm mag} < 0. \tag{3.27}$$

It seems obvious that a sink particle should only be formed out of gas which is gravitationally bound. One can thus call this a necessary condition for gravitational collapse. However, the condition is not sufficient. A gas configuration in virial equilibrium passes this test although it is not collapsing. Furthermore it is not straightforward to define the gravitational binding energy $E_{\rm pot}$ of a gas configuration which is embedded in a cloud of turbulent gas. When we use this check in our comparison tests, we compute the maximum potential inside Ω_i and use this as a reference potential.

3.3.5.2. Jeans instability check

The mass inside the control volume must exceed the local Jeans mass. This is made sure by requiring

$$E_{\rm pot} + 2E_{\rm therm} < 0. \tag{3.28}$$

As the bound state check, this condition represents a necessary but not a sufficient condition for gravitational collapse as it neglects the internal kinetic energy of the gas and it is not clear how to define $E_{\rm pot}$.

3.3.5.3. Potential minimum check

Federrath et al. (2010) introduced this check which has been adopted by other groups in AMR (Gong & Ostriker, 2013) or SPH (Hubber et al., 2013; Wadsley et al., 2011) codes to reduce the formation of spurious sinks. This check allows a sink to be formed only in a cell which hosts a local minimum in the gravitational potential. Although the authors mentioned above find this test important to reduce the production of sinks from transient density fluctuations, it is lacking of a physical justification. A local minimum in the gravitational potential is not a prerequisite for local gravitational collapse. This can be seen in a thought experiment where a constant force field is applied to the region of interest. The addition of a constant force term corresponds to adding a linear term in the gravitational potential. This changes the position and/or existence of local extrema in the potential without changing the local dynamics. This demonstrates why the tidal tensor, which is not affected by the addition of a linear term, is the right quantity for the evaluation of local gravitational collapse (see Section 3.3.2). It is therefore not clear whether the gravitational potential due to pre-existing sinks should be added to the gas potential before applying this check or not[9]. Including the sink potential introduces strong gradients which could wrongfully prevent a sink from being formed by removing or dislocating the potential minimum. On the other hand, the curvature of the potential induced by the sink particles contains the tidal forces that the sink particles exert onto the surrounding gas and should therefore enter the analysis. In our implementation of the potential minimum check we decided to consider only the gravitational potential caused by the gas.

[9] This question only arises when the direct force summation approach is used. When applying the PM method the sink mass is contained in the source term of the Poisson equation and therefore in the resulting potential (see Section 3.5).

3.4. Merging sinks

Sink particles are typically introduced to represent gravitationally collapsed objects whose physical size is orders of magnitude below the grid scale. To decide whether two of those objects are undergoing a merger is therefore beyond the scope of the simulation itself, even in cases where the two sink particles are occupying the exact same cell for a long time. We have to consider physics on a sub-grid scale to decide whether two objects which are close to each other relative to the grid scale will actually get close to each other on the scale of their physical extent. Approaches to sink merging in existing implementations therefore cover a broad spectrum. Krumholz et al. (2004) merge sinks using a FOF algorithm where the linking length is given by the accretion radius of the sink. Formation and subsequent merging of sinks can be seen as one mode of accretion. This merging strategy is clearly targeting young sink particles and the authors mention the possibility to turn off merging at a later stage during the simulation. Wang et al. (2010) and Krumholz et al. (2012) have presented calculations where they use a mass threshold which - once a sink particle has passed it - prevents the sink from being destroyed through merging. Gong & Ostriker (2013) follow a merger friendly strategy as well and merge sink particles as soon as their accretion zones are overlapping. Federrath et al. (2010) have implemented sink merging as an option that can be activated by the user. If switched on, two sink particles will merge whenever their separation is less than one accretion radius, they are converging and they are gravitationally bound to each other.

3.4.1. Merging on a timescale

As Federrath et al. (2010) we share the view that sink merging should be optional in a simulation code since it must be decided based on the very details of the setup and the sub grid physics whether sinks should merge or not. However, in order to bridge the gap between the two extreme cases we present a strategy where we merge sinks based on a collapse timescale. The underlying assumption is that the gas which has just triggered sink formation takes a certain time to collapse to sub grid scale. During this time, the sink represents a "not yet collapsed" object whose size is still comparable to the grid spacing. We therefore merge such a young sink to an "old" one if they are less than one accretion radius apart, or we merge two young sinks if their distance is less than two accretion radii. When we apply this method, we slightly modify the checks for sink creation to be more consistent with the idea of a collapse timescale. In Section 3.3.3 we introduced three timescales of contraction (see Equation 3.26). For sink creation we therefore require the contraction time scale along each direction to be shorter than the chosen time scale of collapse. While this time scale must be adapted to the physical setup considered, the concept was clearly motivated by the lifetime of the first Larson core in simulations of fragmenting turbulent molecular clouds (Larson, 1969).

3.5. Sink particle trajectories

The integration of sink particle motion in different AMR codes mainly differs in the way the sink-sink and sink-gas gravitational forces are computed. A natural approach for a particle mesh code (PM, Hockney & Eastwood (1981)) such as RAMSES is to use the PM method for the sink particles in a similar way as it is used for dark matter particles. Another option is to compute the sink-sink and sink-gas interactions "brute force" by summing up

the pairwise forces directly. Direct summation consists of a loop of size $n_{\rm cells} \times n_{\rm sinks}$ and one of size $n_{\rm sinks}^2$. Simulations involving a large number of sink particles and cells can be slowed down so much that switching to the PM method might be desirable. However, the PM method is not designed for collisional dynamics. We expect it to be inaccurate for situations where the local gravitational field is completely dominated by a sink particle. Federrath et al. (2010) use direct force summation for the sink-sink acceleration and the gas acceleration due to the sink, for the sink acceleration due to the gas they perform "cloud-in-cell" (CIC, Hockney & Eastwood (1981)) interpolation of the gravitational field from the grid values onto the location of the sink. Krumholz et al. (2004) do direct force summation as well, while Gong & Ostriker (2013) use the PM method together with the "triangular-shaped-cloud" (TSC, Hockney & Eastwood (1981)) interpolation scheme. Another distinguishing feature of certain sink particle implementations (Federrath et al., 2010; Krumholz et al., 2004) is the possibility for the sink particles to "sub-cycle" the gas, meaning that multiple sink particle updates are performed within one time step of the computationally much more expensive hydro solver. This technique therefore allows a very small softening length for sink-sink interactions (or no softening at all) which pushes the resolution of the sink-sink forces beyond the grid spacing.

3.5.1. PM method

Our implementation of the PM method for sink particles makes use of the PM method for dark matter particles already present in RAMSES (Teyssier, 2002). Each sink particles mass is distributed equally onto a spherical "swarm" of equally spaced RAMSES particles. The spacing of these particles is half the grid spacing, the radius of the sphere is a free parameter and sets the gravitational softening length. The mass of each particle is deposited onto the grid using the CIC scheme with cloud size being equal to the local grid spacing. This can be seen as a "fuzzy" top hat softening. The Poisson equation is solved using one of the solvers implemented in RAMSES (multigrid: Guillet & Teyssier (2011), conjugate gradient: Teyssier (2002)) and the gravitational field is computed using the 5 point finite difference approximation. The gravitational acceleration of each swarm particle is obtained by CIC interpolation from the cell center values. Finally, averaging over all particles belonging to one sink yields the acceleration of the sink.

3.5.2. Direct force summation

When doing direct force summation, only the gas density is considered as source term for the Poisson equation. Accelerations due to sink-sink and sink-gas interactions are computed by looping over all pairwise combinations and computing their mutual attraction.[10] We apply a Plummer softening (Aarseth, 1963).

$$\boldsymbol{F}(\boldsymbol{r}) = -\boldsymbol{r}\frac{GM}{(|\boldsymbol{r}|^2 + R_{\rm soft}^2)^{3/2}} \tag{3.29}$$

to both, the sink-sink and the sink-gas forces where the softening length is a free parameter. As Krumholz et al. (2004) point out, the gravitational force should not be reduced too much due to the softening at the boundary of the sink accretion zone. We therefore set the softening radius to half the accretion radius as a default. This implies that the resolution

[10] All the gas in one cell is assigned to the cell center location for this step.

of the sink-sink forces is of the order of the grid spacing. We are therefore for instance not able to follow two sinks orbiting each other inside one cell.

3.5.3. The integrator

In RAMSES particles are integrated using a second order midpoint scheme which - for constant time steps - is equivalent to the classical leapfrog method (Teyssier, 2002). We apply the same method to the sink particles. Since we use identical softening for sink-sink as for sink-gas forces, the maximum accelerations of gas and sink particles are comparable. We therefore update the sink particles using the same time step as for the gas at the finest level of refinement.[11] In RAMSES calculations the minimum free fall time occurring has to be resolved,

$$\Delta t < C\sqrt{\frac{3\pi}{32G\rho_{\max}}}, \tag{3.30}$$

where $0 < C < 1$ is a constant (Teyssier, 2002). When using the PM method, the maximum density $\rho_{\max}$ is identified after the particle mass deposition through the CIC scheme. In case of direct force summation, as soon as the maximum sink density obtained from the Plummer density distribution $\rho_{\rm Plummer} = 3M_{\rm sink}/4\pi r_{\rm soft}^3$ exceeds the maximum gas density, $\rho_{\rm Plummer}$ is used for computing the time step through Equation 3.30. Furthermore, sink particles like any other particle in RAMSES are allowed to travel only a fraction of the local mesh spacing within one time step. As a last sink related restriction on the time step, we set the condition that only a fraction of the available gas can be accreted within one time step (see Section 3.6.3).

3.6. Accretion onto sinks

After its formation, a sink particle accretes gas from nearby cells. Different methods to perform accretion have been described and justified using various tests. However, direct comparisons of results obtained by different accretion schemes have not been performed. We implemented and compared three different modes of accretion. Fixed threshold accretion (TA), Bondi-Hoyle accretion (BH) and what we call flux accretion (FA) where the accretion rate is computed based on the mass flux rate into the sink accretion zone. In the following subsections we briefly describe the different schemes. In all schemes, velocity and position of the accreted gas relative to the sink are used to update position and velocity of the sink as well as to keep track of the angular momentum that has been removed from the gas by the sink particle.

[11] RAMSES allows a finer level in the AMR hierarchy to "sub-cycle" a coarser level by updating the finer level twice while the coarse level is updated only once.

$$M_s^{\rm new} = M_s^{\rm old} + \sum_{i \in cells} \Delta m_i \tag{3.31}$$

$$\boldsymbol{R}_s^{\rm new} = (\boldsymbol{R}_s^{\rm old} M_s^{\rm old} + \sum_{i \in cells} \boldsymbol{r_i} \Delta m_i)/M_s^{\rm new} \tag{3.32}$$

$$\boldsymbol{V}_s^{\rm new} = (\boldsymbol{V}_s^{\rm old} M_s^{\rm old} + \sum_{i \in cells} \boldsymbol{v_i} \Delta m_i)/M_s^{\rm new} \tag{3.33}$$

$$\begin{aligned} \boldsymbol{L}_s^{\rm new} = {} & \boldsymbol{L}_s^{\rm old} + (\boldsymbol{R}_s^{\rm new} - \boldsymbol{R}_s^{\rm old}) \times (\boldsymbol{V}_s^{\rm new} - \boldsymbol{V}_s^{\rm old}) M_s^{\rm old} \\ & + \sum_{i \in cells} (\boldsymbol{R}_s^{\rm new} - \boldsymbol{r}_i) \times (\boldsymbol{V}_s^{\rm new} - \boldsymbol{v}_i) \Delta m_i \end{aligned} \tag{3.34}$$

3.6.1. Threshold accretion (TA)

Federrath et al. (2010) use this method where gas is accreted from cells which are closer than $R_{\rm acc}$ to an existing sink and whose density exceeds the threshold $\rho_{\rm sink}$. Additionally, the gas in a cell is required to be bound to the sink and the radial component of the gas velocity relative to the sink needs to be negative. If these conditions are met, the accreted gas mass from a cell is

$$\Delta m_i = \max(0.5(\rho - \rho_{\rm sink})(\Delta x)^3, 0), \tag{3.35}$$

where Δx is the size of the cell. In sorting the sink particles by mass we ensure that the the most massive sink gets most of the mass in case of multiple sinks accreting from the same cell. Federrath et al. (2010) improve this by checking which sink the gas is bound to the strongest.

3.6.2. Bondi-Hoyle accretion (BH)

Krumholz et al. (2004) compute the sink accretion rates based on the theory by Bondi, Hoyle and Littleton (Bondi, 1952; Hoyle & Lyttleton, 1939). The Bondi-Hoyle radius is

$$r_{\rm BH} = \frac{GM_\star}{(v_\infty^2 + c_\infty^2)} \tag{3.36}$$

and the corresponding accretion rate is given by

$$\dot{M}_{\rm BH} = 4\pi \rho_\infty r_{\rm BH}^2 \sqrt{\lambda^2 c_\infty^2 + v_\infty^2}, \tag{3.37}$$

where $M_\star$ is the mass of the star and v_∞, c_∞, ρ_∞ are the velocity of the gas relative to the star, the sound speed and the density far from the star relatively. The parameter λ depends on the equation of state, $\exp(3/2)/4 \approx 1.12$ is the correct value for isothermal gas. When computing the sink accretion rate, we replace $M_\star$ by the sum of the sink mass and the gas mass inside the sink radius to increase the accretion rate of very low mass sinks. Using the recipe given by Krumholz, we choose v_∞, c_∞ to be the values at the sink location and we extrapolate from the weighted mean density inside the sink accretion radius $\overline{\rho}$ to

$$\rho_\infty = \frac{\overline{\rho}}{\alpha(\overline{r}/r_{\rm BH})} \tag{3.38}$$

with $\alpha(x) \equiv \rho(x)/\rho_\infty$ being the density profile that arises from the transsonic solution of the spherical Bondi problem as a function of the dimensionless radius $x \equiv r/r_{\rm BH}$. The radius $\overline{r}$ corresponding to the density $\overline{\rho}$ is chosen to match expected results. To average the density inside the sink radius and to smoothen accretion when the sink particle is moving through the grid, we use the same kernel function as Krumholz et al. (2004) which assigns every cell inside the accretion zone a weight

$$w \propto \exp(-r^2/r_k^2). \tag{3.39}$$

Note that in contrast to the description given by Krumholz et al. (2004), we simply fix $\overline{r}$ in Equation 3.38 as well as the kernel size r_k to half the accretion radius. In the presence of rotational flows around the sink, the Bondi-Hoyle accretion rate is an overestimation of the effective accretion rate. We use trick by Krumholz et al. (2004) to reduce the accretion rate: A cell inside the accretion radius is divided into 8^3 little sub-cubes. Using the specific energy and the specific angular momentum of the gas, the "closest approach" of each cube to the sink particle is estimated assuming ballistic trajectories.The number of cubes that will not make it closer to the sink than $0.25\Delta x$ is counted and the Bondi-Hoyle accretion rate is reduced by the corresponding factor.

3.6.3. Flux accretion (FA)

In this accretion method we set the accretion rate equal to the mass flux rate into the sink accretion zone. Gong & Ostriker (2013) first describe this using the fluxes at the cell boundaries returned by the Riemann solver. Since these fluxes are relative to the grid they need to be corrected for the sink motion when a sink particle moves through a density gradient. We therefore take a slightly different approach and compute the mass flux into the accretion zone $\Omega_{\rm acc}$ using Gauss' divergence theorem,

$$\dot{M}_{\rm flux} = -\int_{\Omega_{\rm acc}} \mathrm{div}\big(\rho(\boldsymbol{v} - \boldsymbol{v}_{\rm sink})\big)\mathrm{d}V. \tag{3.40}$$

As we do not allow for negative accretion rates, the gas mass inside the accretion zone can only decrease. To keep the gas density inside the accretion zone close to the sink threshold density in the long term, we correct this mass flux rate by a small factor and use the following "flux accretion rate"

$$\dot{M}_{\rm FA} = \left[1 + 0.1\lg\left(\frac{\overline{\rho}}{\rho_{\rm sink}}\right)\right]\dot{M}_{\rm flux}, \tag{3.41}$$

where $\overline{\rho}$ is the mean gas density inside the accretion zone and $\rho_{\rm sink}$ is the user-defined sink threshold. We compute the gas mass that is removed from a cell Δm_i in the accretion zone in a mass weighted fashion,

$$\Delta m_i = \begin{cases} \Delta t \frac{\dot{M}_{\rm FA}}{n_{\rm cells}} \frac{\rho_i}{\overline{\rho}} & \text{if} \quad \dot{M}_{\rm FA} \geq 0, \\ 0 & \text{if} \quad \dot{M}_{\rm FA} < 0, \end{cases} \tag{3.42}$$

where $n_{\rm cells}$ is the number of cells in the accretion zone. Since in FA accretion we remove gas from the individual cells in a mass weighted fashion, the gas inside each cell is reduced by the same factor. We make use of this fact to define a new time step criterion to ensure

that no cell is emptied completely rather than artificially capping accretion. We compute the total available gas mass inside the accretion zone $M_{\rm gas}$ and require

$$\Delta t_{\rm acc} < C\frac{M_{\rm gas}}{\dot{M}_{\rm FA}}, \tag{3.43}$$

where we set $C = 0.75$ as a default. Using this time step constraint makes sure that not more than 75 per cent of the gas is removed from one cell within a single time step.

3.6.4. "No-L" accretion

When sink particles accrete gas they remove angular momentum from the simulation. A sink represents a collapsed object which is much smaller then the grid spacing. It is therefore unphysical to simply assign the accreted angular momentum to physical object the sink represents since it would very quickly be spinning at unrealistically high rates. The sink particle therefore acts as a sink not only for the mass, but also for angular momentum. This facilitates accretion from disk-like structures by removing the necessity to transport angular momentum outwards. This was highlighted and found to be important in SPH simulations by Hubber et al. (2013). They solve this problem by feeding back to the gas the angular momentum that has been accreted previously. We use an approach described by Krumholz et al. (2004): We decompose the momentum in the motion of the gas relative to the sink into a radial and a tangential part. While the radial part of the momentum is transferred to the sink, the tangential part is assigned to the remaining gas. This corresponds to an acceleration of the remaining gas in the tangential direction since the momentum in the tangential motion remains constant while the gas mass decreases. We keep this "no-L accretion" optional for all accretion schemes. Note that this method does only work if a sink is accreting directly from the gas. In the case where sink formation and subsequent merging work as an accretion mechanism, this technique fails as angular momentum is removed whenever sinks merge.

3.7. Tests

In this section we report the tests that we have performed using different sink particle implementations. We describe tests on sink formation, sink merging and accretion onto sinks in this order. In the Appendix we discuss two small test cases that concern the integration of the sink trajectories. We try to separate those tests as far as possible which means for example, that when comparing different methods for sink formation, all codes use the same accretion recipe[12]. We test the creation of sink particles using a Boss & Bodenheimer test (BB test, Boss & Bodenheimer (1979)) and fragmentation in turbulent molecular gas. We compare three different algorithms for sink formation: a cell-based, for which we use the acronym CELL, a peak-based (acronym PEAK) and a clump-based (acronym CLUMP) strategy.

1. In the CELL approach, a sink is formed in every cell that crosses the sink formation threshold $\rho_{\rm sink}$. Thereby, the gas exceeding the threshold is immediately absorbed by the sink. Sinks are merged using the FOF technique where we have chosen the accretion radius as linking length.

[12]This is not always possible, especially since sink formation and merging as it is described by Krumholz et al. (2004) blurs the line between sink formation and accretion.

2. The PEAK strategy discretizes the computational domain by considering every local density peak above $\rho_{\rm sink}$ for sink formation. A sphere with the size of the accretion radius is defined around the density peak and used as integration domain to compute contraction rates, and energies. The gas inside such a sphere must pass the proximity check, Jeans instability check, bound state check, collapse check and potential minimum check to trigger sink formation.

3. Our new sink formation algorithm is denoted as CLUMP approach. It allows sinks to be formed only at the density peaks above $\rho_{\rm sink}$ of clumps having a high enough peak-to-saddle ratio. The gas surrounding those peaks is then subjected to the collapse check, proximity check and the virial check.

See Section 3.3 for a more detailed description of the different checks mentioned above. The turbulent setup is used to compare sink merging. For sink accretion we consider two test cases, spherical Bondi accretion and accretion from a disk. Those two test cases are applied to the different accretion schemes described in Section 3.6. We compare Bondi-Hoyle accretion (BH), flux accretion (FA), threshold accretion(TA) and threshold accretion with a threshold reduced by a factor of 10 (TA-low).

3.7.1. Boss & Bodenheimer test

We performed a series of tests where we followed the collapse and fragmentation of a rotating core, known as the Boss & Bodenheimer (BB) test. This test consists of a gas sphere in solid body rotation which is seeded with a $m = 2$ density perturbation. The sphere collapses into one or more fragments, depending on the parameters used in the setup, and most importantly, on the sink particle algorithm used. BB tests have been used extensively by many authors to test fragmentation in hydrodynamical codes in general and perform resolution studies and code comparisons (e.g., Bate & Burkert, 1997; Boss & Bodenheimer, 1979; Commerçon et al., 2008; Truelove et al., 1997). We choose the same initial conditions as Federrath et al. (2010) when they tested their sink particle algorithm. The parameters of the setup are shown in Table 3.1. The threshold density for sink formation is chosen as the density above which the local Jeans length is not resolved by 4 cells anymore. A cell is refined when the local Jeans length is less than $4\Delta x$. We use the flux accretion scheme for this test and the sink accelerations are computed using direct force summation.

3.7.1.1. Isothermal EOS

For isothermal gas in the absence of magnetic fields, the initial $m = 2$ perturbation collapses and forms a filament. No matter what resolution is chosen, this filament will eventually become dense enough to violate the Truelove criterion and fragment artificially (Truelove et al., 1997). This can be observed in Figure 3.9 which shows a snapshot for the setup specified in Table 3.1 but including 4 additional levels of refinement, setting the minimum cell size to $\Delta x_{\rm min} = 0.4\,{\rm au}$. Artificial fragmentation is clearly visible.

The filamentary nature of the collapse makes the isothermal BB test a "worst case" scenario for sink formation. Forming sinks in a filament or a sheet will always introduce an artificial length scale which corresponds to the typical sink spacing. However, knowing the behavior of the sink formation algorithm when applied to a collapsing filament is relevant since we know from previous simulations of supersonic turbulence (e.g., Collins et al.,

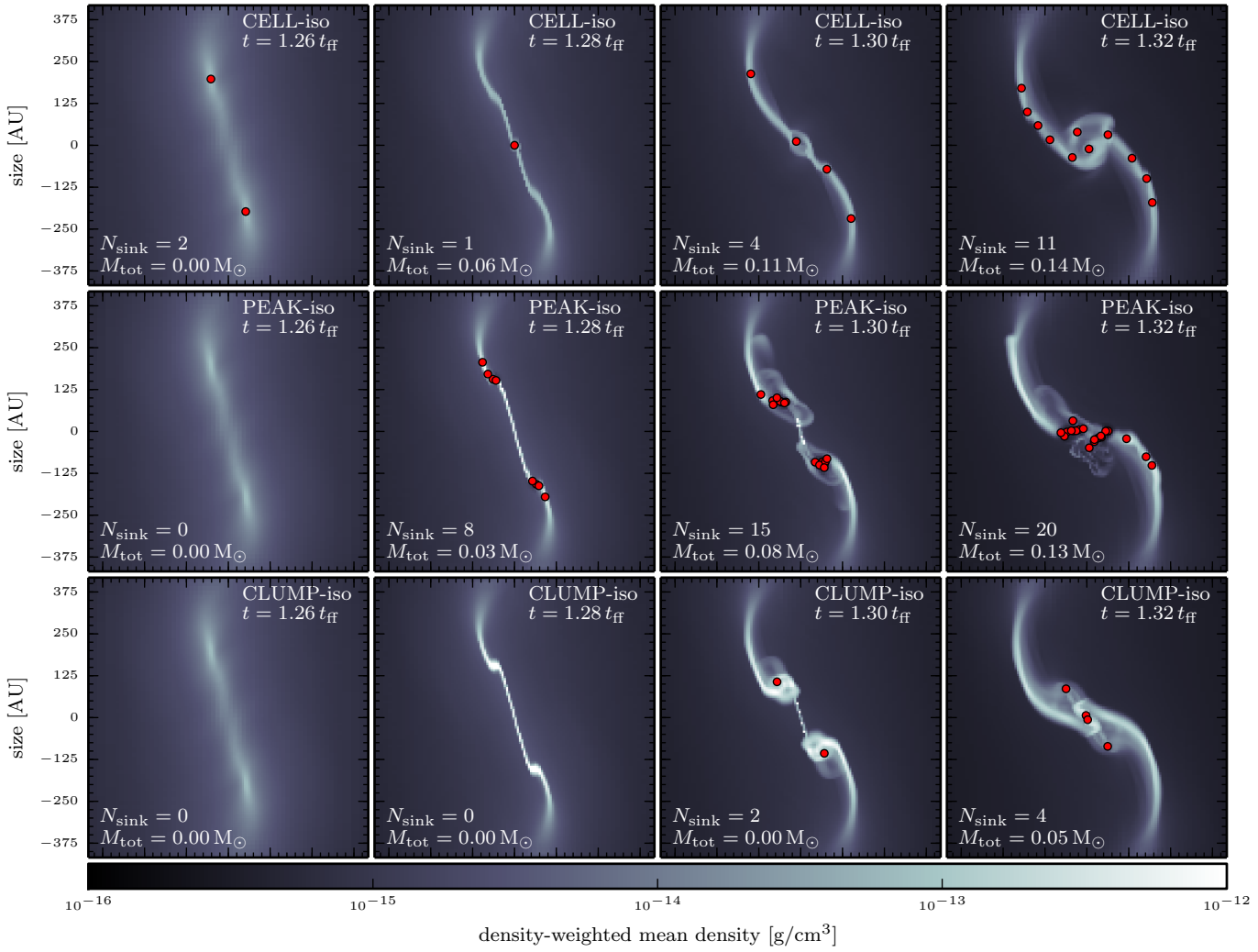


Figure 3.7: Comparison of different sink formation algorithms on an isothermal Boss & Bodenheimer test. The time when each snapshot was taken is given in terms of $t_{\rm ff} \approx 34.1\,{\rm kyr}$. Sink particles are marked with red dots and the size of the dots corresponds to the sink accretion radius. The cell-based algorithm (top row) successfully prevents violation of the Truelove criterion by forming sinks in all cells that cross the density threshold. During the subsequent evolution, constantly ongoing sink formation and merging act as an effective way of accretion and lead to roughly equally spaced sinks along the filament. The peak-based method (middle row) forms 20 sinks from artificial fragments while our new clump-based algorithm (bottom row) allows only 4 of the artificial fragments to trigger formation of a sink during the course of our experiment.

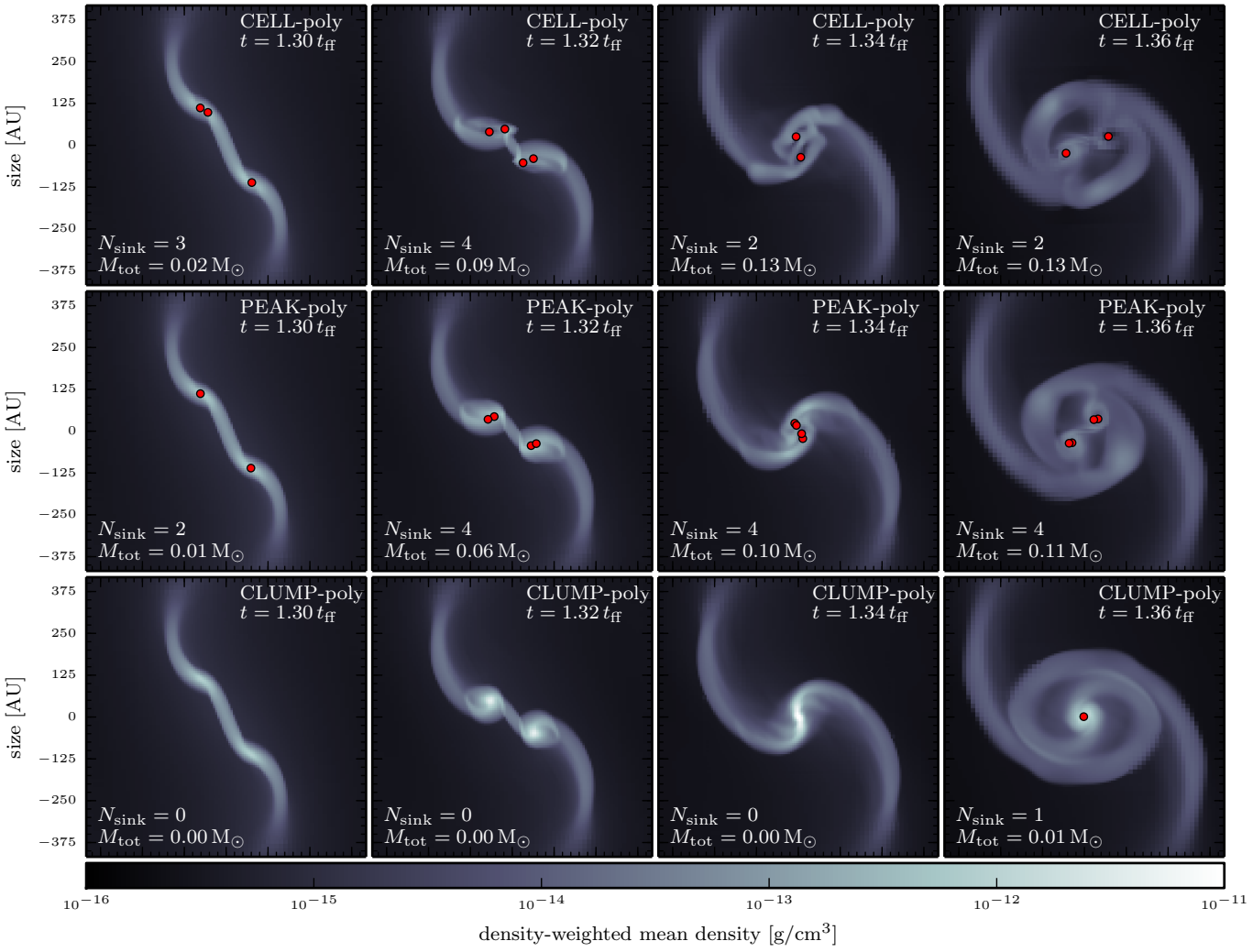


Figure 3.8: Same as Figure 3.7 when using the piecewise polytropic EOS 3.44. The heating causes the filament to form two distinct fragments. The cell-based method (top row) forms multiple sinks in both fragments that later merge into two sinks forming a binary system. The peak-based algorithm (middle row) triggers formation of a very tight binary inside each fragment. These two binaries orbit each other on a trajectory similar to the one observed in the run where the cell-based sink formation criteria are used. The clump-based method (bottom row) does not allow those two fragments to form a sink as they have too much rotational (the second panel from the left shows that each fragment is in fact a small disk-like structure) and thermal support. Only after the two fragments collide, enough low angular momentum gas is left in the center to form a sink.

Table 3.1: Simulation parameters for the Boss & Bodenheimer test.

Radius	$R = 5.0 \times 10^{16}\,\mathrm{cm} \approx 3300\,\mathrm{au}$
Mass	$M = 1\,\mathrm{M}_\odot$
Average density	$\rho_0 = 3.82 \times 10^{-18}\,\mathrm{g/cm^3}$
Free-fall time	$t_\mathrm{ff} = 1.075 \times 10^{12}\,\mathrm{s} \approx 34\,\mathrm{kyr}$
Density perturbation	$\rho(\phi) = \rho_0(1 + 0.1\cos(2\phi))$
Isothermal sound speed	$c_s = 1.66 \times 10^4\,\mathrm{cm/s}$
$E_\mathrm{therm}/E_\mathrm{grav}$	$\alpha = 0.26$
Angular velocity	$\Omega = 7.2 \times 10^{-13}\,\mathrm{s^{-1}}$
$E_\mathrm{rot}/E_\mathrm{grav}$	$\beta = 0.18$
Box size	$L_\mathrm{box} = 2.0 \times 10^{17}\,\mathrm{cm}$
Cell size at levelmax	$\Delta x_\mathrm{min} = 6.5\,\mathrm{au}$
Sink radius	$R_\mathrm{acc} = 4\Delta x_\mathrm{min} \approx 26\,\mathrm{au}$
Sink density threshold	$\rho_\mathrm{sink} = \begin{cases} 8.5 \times 10^{-14}\,\mathrm{g/cm^3} & \text{(iso)} \\ 5.5 \times 10^{-13}\,\mathrm{g/cm^3} & \text{(poly)} \end{cases}$

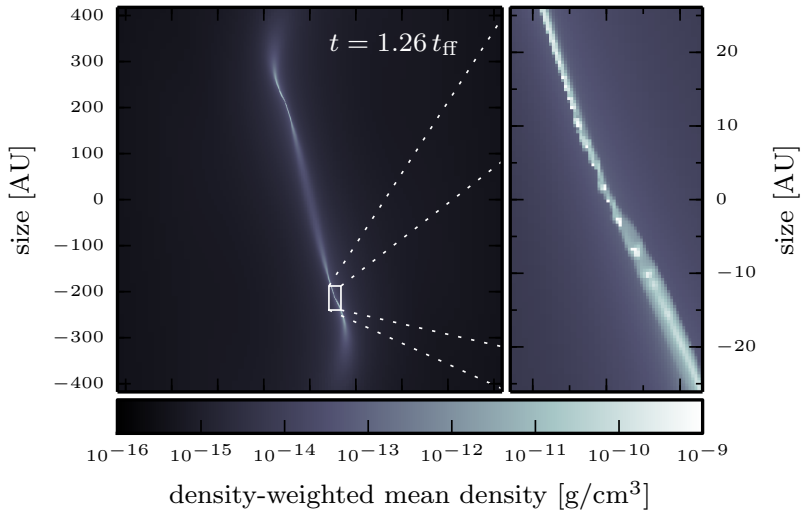


Figure 3.9: Zoom snapshot of a high resolution isothermal Boss & Bodenheimer test showing artificial fragmentation.

2012; Heitsch et al., 2008; Klessen et al., 2004) that gas tends to assemble filaments. From observations we know that filaments are ubiquitous in star-forming clouds (André et al., 2010).

Figure 3.7 shows that our 3 methods for sink formation lead to very different results. The CELL algorithm successfully prevents a violation of the Truelove criterion by immediately absorbing gas that exceeds the density threshold into sinks. The ongoing process of sink creation, accretion and merging results in roughly equally spaced sinks along the filament. The spacing is determined by the resolution dependent sink accretion radius which acts as linking length in the FOF algorithm. As mass is accreted from the continuous 1-dimensional filament onto the discrete number of sinks, the filament is effectively fragmenting on a resolution dependent scale, very similar to the artificial fragmentation in the Truelove et al. (1997) sense. In contrast, the PEAK as well as the CLUMP method do not form a sink until the filament has fragmented artificially. While the PEAK scheme triggers sink formation in almost every artificial fragment, the CLUMP approach is more restrictive and allows only 4 sinks to form before we stop the experiment. The clump finder together with the virial check can prevent most of the artificial fragments from forming a sink. Only those artificial fragments which are dominating the local gravitational field will trigger formation of a sink. Note that sink formation in all 3 cases is still ongoing after the last snapshot shown in Figure 3.7.

3.7.1.2. Piecewise polytropic EOS

Heating the gas is a possible way to prevent the filamentary "catastrophy" described in the last section. We thus repeat the test introducing the same piecewise polytropic equation of state (EOS) as Federrath et al. (2010),

$$P = \begin{cases} {c_s}^2 \rho & \text{if} \quad \rho \leq 2.5 \times 10^{-16} \frac{\text{g}}{\text{cm}^3}, \\ \kappa_1 \rho^{1.1} & \text{if} \quad 2.5 \times 10^{-16} \frac{\text{g}}{\text{cm}^3} < \rho \leq 5.0 \times 10^{-15} \frac{\text{g}}{\text{cm}^3}, \\ \kappa_2 \rho^{4/3} & \text{if} \quad 5.0 \times 10^{-15} \frac{\text{g}}{\text{cm}^3} \leq \rho, \end{cases} \tag{3.44}$$

where the values κ_1 and κ_2 are chosen such that P is a continuous function of ρ. When using this EOS the heating slows down the collapse onto the filament and causes the formation of a well defined fragment at each end of the filament (see Figure 3.8). The CELL run forms and merges sinks in both fragments leading to two sinks forming a binary system. The PEAK run triggers formation of two sinks in each fragment. Note that we do not allow sinks to merge when using the PEAK method for sink formation. When sink merging is turned on, the two sinks inside each fragment merge quickly after the formation of the second sink and the subsequent evolution is very close to the one seen in the CELL run. The CLUMP method identifies the density peak inside each fragment as a possible location for sink formation. Yet both of the fragments fail the virial check due to a combination of rotational and thermal support. Note that although the Truelove criterion is violated and the local Jeans length is not resolved by 4 cells inside the fragments, there is no artificial fragmentation happening. At $t = 1.34 t_{\text{ff}}$ the two initial fragments undergo a grazing collision leading to ejection of some high angular momentum gas and one fragment in the center which then forms a single sink.

At this place we want to add a note on the issue of numerical convergence. The isothermal setup is scale free and the fragmentation scale is therefore determined by the artificial fragmentation at the grid scale. More generally, Martel et al. (2006) showed that the fragmentation scale is resolution dependent for isothermal SPH simulations. Consequently,

there is no numerical convergence for the isothermal case. The piecewise polytropic case deserves a little more attention. The "knee" in the EOS introduces a physical scale that determines the properties of the resulting fragments (Larson, 2005). It seems therefore possible that, once the fragmentation scale is properly resolved, changes in the resolution will not change the results of the numerical experiment anymore. We have thus performed a convergence study on the piecewise polytropic setup where we have increased the sink density threshold according to the numerical resolution. We found that the results for all three sink formation algorithms to be *not* converged in this sense. To understand this behavior one can consider the case of a polytropic index of 5/3. In this case, the heating is so strong that the increasing pressure will eventually stop the collapse of the fragments, resulting in a stable hydrostatic configuration. Increasing the sink density threshold will therefore at some point prevent sink formation completely. In the case of a polytropic index of 4/3 there is no stable polytrope (Bonnor, 1958) and every fragment must collapse eventually. However, we found that by increasing the sink threshold density, one can arbitrarily delay the moment when this threshold is crossed. This is critical as the fragments are in violent dynamical interactions while they are contracting. Delaying the moment of sink formation will therefore alter the results and prevent convergence. It is therefore the physical setup itself which is not converging. One situation where we can imagine convergence in the above sense is the isothermal collapse of a spherical gas configuration as it is probably the case when resolving the second core collapse. Another way to approach the issue of numerical convergence in the presence of sinks is by arguing that the sink density threshold is a physical rather than a numerical parameter and therefore kept fixed as the resolution increases. We do believe that this type of numerical convergence can be achieved for the above setup. Explicitly demonstrating this type of convergence is beyond the scope of this paper.

3.7.2. Collapse of a turbulent molecular cloud

Sink particles are an essential ingredient of simulations that model the formation of a star cluster inside molecular gas (e.g., Bate, 2012; Girichidis et al., 2011; Krumholz et al., 2012). We use such a scenario to compare the different sink formation methods. We use two setups as similar as possible to the top hat runs in Girichidis et al. (2011). An isothermal, initially spherical gas configuration is seeded with turbulent motions that decay, allowing the cloud[13] to collapse and fragment. Some physical and numerical parameters for this test are summarized in Table 3.2. The velocity field is modeled by Burgers turbulence ($P(k) \propto k^{-4}$) which is in agreement with measured size-linewidth relations in molecular clouds (Heyer et al., 2009; Larson, 1981). We use mixed turbulence which means that the initial velocity field contains solenoidal (divergence free) as well as compressive (curl free) modes. The sink formation threshold $\rho_{\rm sink}$ is chosen such that the Jeans length at this density is resolved by exactly 4 cells at the finest level. We use a mass based variant of the Jeans refinement criterion which guarantees that the *smallest* Jeans mass in the calculation is resolved by a fixed number of cells throughout the whole calculation. We therefore compute the mass in one cell at the maximum density $\rho_{\rm sink}$ and use this as a mass resolution element. During the calculation, a cell is refined as soon as its gas mass exceeds the mass resolution element. This leads to a roughly constant number of cells

[13] An object of $100\,{\rm M}_\odot$ would usually by considered a "clump" inside a molecular cloud rather than a "cloud" itself. We label it as "cloud" because we use the word clump already for a much smaller structure in the context of sink formation.

Table 3.2: Physical and numerical parameters for the collapsing molecular cloud test. The two setups are generated using different random number seeds (s1/s2) and slightly different normalizations of the velocity field.

Radius	$R = 3.0 \times 10^{17}\,\mathrm{cm} \approx 0.01\,\mathrm{pc}$
Mass	$M = 100\,\mathrm{M}_\odot$
Density	$\rho = 1.76 \times 10^{-18}\,\mathrm{g/cm^3}$
Free-fall time	$t_\mathrm{ff} = 5.0 \times 10^4\,\mathrm{yr}$
Mean molecular weight	$\mu = 2.3$
Temperature	$T = 20\,\mathrm{K}$
Isothermal sound speed	$c_s = 2.68 \times 10^4\,\mathrm{cm/s}$
Sound crossing time	$t_\mathrm{sound} = 7.1 \times 10^5\,\mathrm{yr}$
$E_\mathrm{therm}/E_\mathrm{grav}$	$\alpha = 0.04$
Turbulent mach number	$\mathcal{M}_\mathrm{rms} = 3.65$ / 3.33
Turbulent crossing time	$t_\mathrm{turb} = 1.9 \times 10^5\,\mathrm{yr}$ / $2.1 \times 10^5\,\mathrm{yr}$
$E_\mathrm{kin}/E_\mathrm{grav}$	$\beta = 0.18$ / 0.15
$E_\mathrm{solenoidal}/E_\mathrm{compressive}$	$\gamma = \underbrace{1.82}_{s1}$ / $\underbrace{1.53}_{s2}$
Box size	$L_\mathrm{box} = 1.60 \times 10^{18}\,\mathrm{cm}$
max level of refinement	$l_\mathrm{max} = 13$
min level of refinement	$l_\mathrm{min} = 8$
Cell size at levelmax	$\Delta x_\mathrm{min} = 13.05\,\mathrm{au}$
Sink accretion radius	$R_\mathrm{acc} = 3\Delta x_\mathrm{min}$
Sink softening	$R_\mathrm{soft} = 1.5\Delta x_\mathrm{min}$
Sink threshold	$\rho_\mathrm{sink} = 2.46 \times 10^{-14}\,\mathrm{g/cm^3}$
Mass resolution element	$m_\mathrm{res} = \Delta x_\mathrm{min}^3 \rho_\mathrm{sink} \approx 10^{-4}\mathrm{M}_\odot$

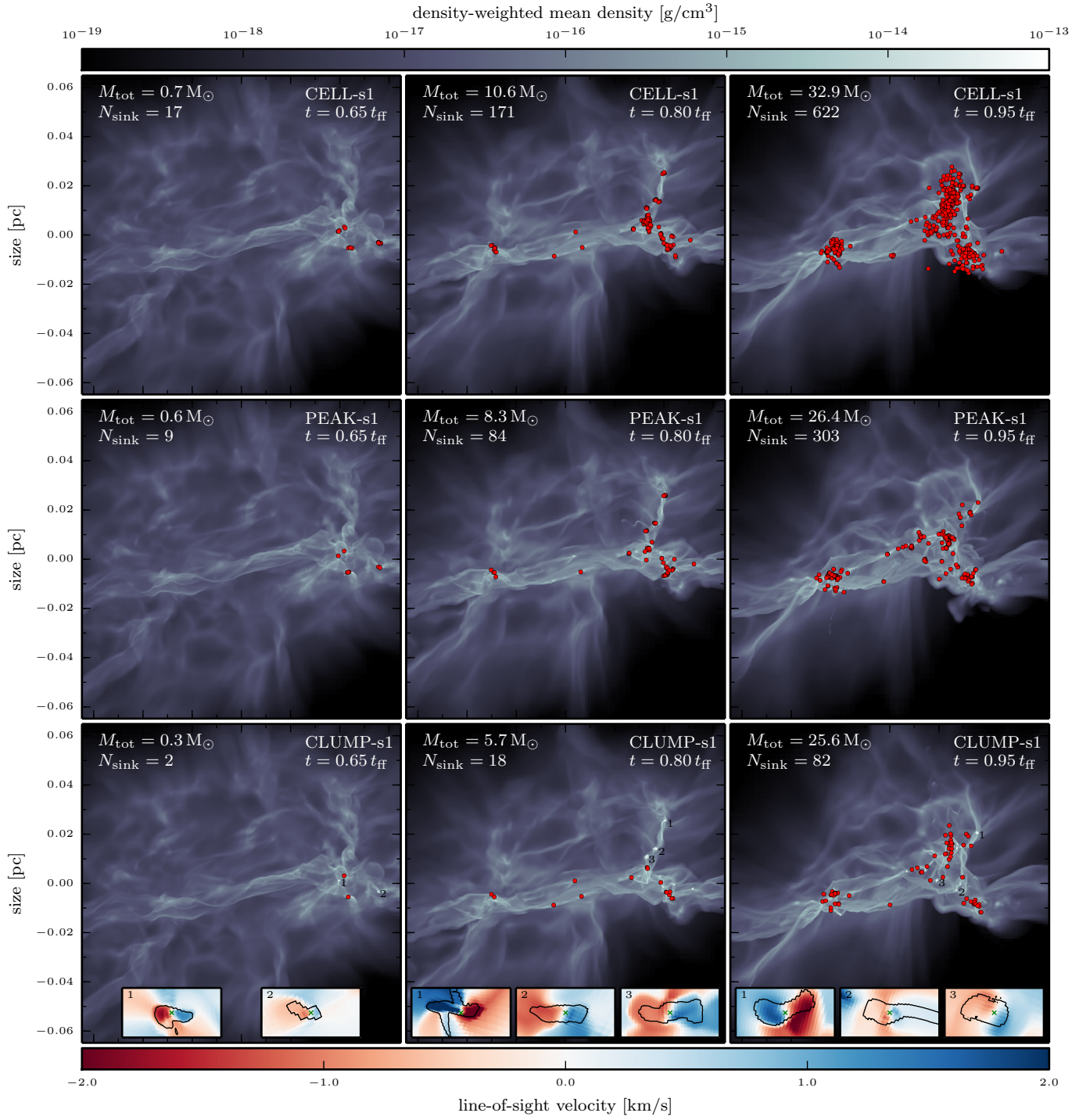

density-weighted mean density [g/cm^3]
10^{-19}
10^{-18}
10^{-17}
10^{-16}
10^{-15}
10^{-14}
10^{-13}
$M_{\rm tot} = 0.7\,{\rm M}_\odot$
$N_{\rm sink} = 17$
CELL-s1
$t = 0.65\,t_{\rm ff}$
$M_{\rm tot} = 10.6\,{\rm M}_\odot$
$N_{\rm sink} = 171$
CELL-s1
$t = 0.80\,t_{\rm ff}$
$M_{\rm tot} = 32.9\,{\rm M}_\odot$
$N_{\rm sink} = 622$
CELL-s1
$t = 0.95\,t_{\rm ff}$
$M_{\rm tot} = 0.6\,{\rm M}_\odot$
$N_{\rm sink} = 9$
PEAK-s1
$t = 0.65\,t_{\rm ff}$
$M_{\rm tot} = 8.3\,{\rm M}_\odot$
$N_{\rm sink} = 84$
PEAK-s1
$t = 0.80\,t_{\rm ff}$
$M_{\rm tot} = 26.4\,{\rm M}_\odot$
$N_{\rm sink} = 303$
PEAK-s1
$t = 0.95\,t_{\rm ff}$
$M_{\rm tot} = 0.3\,{\rm M}_\odot$
$N_{\rm sink} = 2$
CLUMP-s1
$t = 0.65\,t_{\rm ff}$
$M_{\rm tot} = 5.7\,{\rm M}_\odot$
$N_{\rm sink} = 18$
CLUMP-s1
$t = 0.80\,t_{\rm ff}$
$M_{\rm tot} = 25.6\,{\rm M}_\odot$
$N_{\rm sink} = 82$
CLUMP-s1
$t = 0.95\,t_{\rm ff}$
size [pc]
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−2.0
−1.0
0.0
1.0
2.0
line-of-sight velocity [km/s]

Figure 3.10

Figure 3.10: Snapshots comparing the evolution and sink formation of the s1 run for the different sink formation algorithms. The number of sinks and the total mass in sinks is indicated in each snapshot. Sink particles are marked as red dots where the size of the dots is exaggerated in order to be visible and thus not to scale with the rest of the image. The top row shows the results for the cell-based, the middle row for the peak-based and the bottom row for our new clump-based algorithm. The little inlets in the bottom row show enlargements of the most prominent regions that have not yet triggered sink formation by the CLUMP algorithm. The regions are indicated with a little number in the corresponding snapshot. Each inlet covers 500 au × 250 au in size and shows a cut plane through the density peak which is oriented along the angular momentum of the gas surrounding the peak. The black line shows the density contour at $\rho_{\rm contour} = 1.0 \times 10^{-14}\,{\rm g/cm^3}$ and the color indicates the velocity component perpendicular to that plane. The inlets thus show that the densest sink-less regions are little disks that have considerable rotational support. Therefore these disks fail the virial check and form no sink as they are not undergoing gravitational collapse.

($\gtrsim 10^6$) resolving the collapsing cloud throughout the whole calculation and prevents the code from de-refining to low levels early in the calculation when the Jeans length is still large.

We applied each of the 3 methods for sink formation (CELL/PEAK/CLUMP) to both setups (s1/s2) leading to a total of 6 runs. The s1 runs are stopped at $t = 0.95 t_{\rm ff}$ and the s2 runs at $t = 0.85 t_{\rm ff}$. By this time a total mass of $> 20\,{\rm M}_\odot$ has assembled in sinks in each run corresponding to a star formation efficiency of > 20 per cent. We use the same accretion scheme (FA, no-L accretion, see Section 3.6) for all runs. The sinks accelerations are computed as direct sums (see Section 3.5). Figures 3.10 and 3.11 show the temporal evolution of the cloud and the sinks formed by each of the three sink formation algorithms for the two setups. The large scale evolution of the cloud is barely affected by the differences in the sink algorithms but the small scale structure of the gas as well as number and properties of the sinks formed do differ. The most remarkable property seen in these snapshots is the high density regions marked with little numbers in the bottom row. These regions are relatively dense and massive but the CLUMP algorithm has not formed a sink at the time the snapshot was taken. Closer inspection of those regions yields strong vorticity in the velocity field and an internal kinetic energy which is $\approx \frac{1}{2} E_{\rm grav}$ which causes the virial check to prevent sink formation. The fact that those dense regions are actually little disks is shown in the inlets in the bottom row of Figure 3.10.

In Figure 3.12 we plot the number of sinks and the total mass in sinks as a function of time for the 6 runs. It is apparent that the total mass in sinks mainly depends on the initial conditions while the details of the sink formation algorithm have a strong effect on the number of sinks formed. Table 3.3 contains some statistical properties of the sink particle distribution at the end of each run.

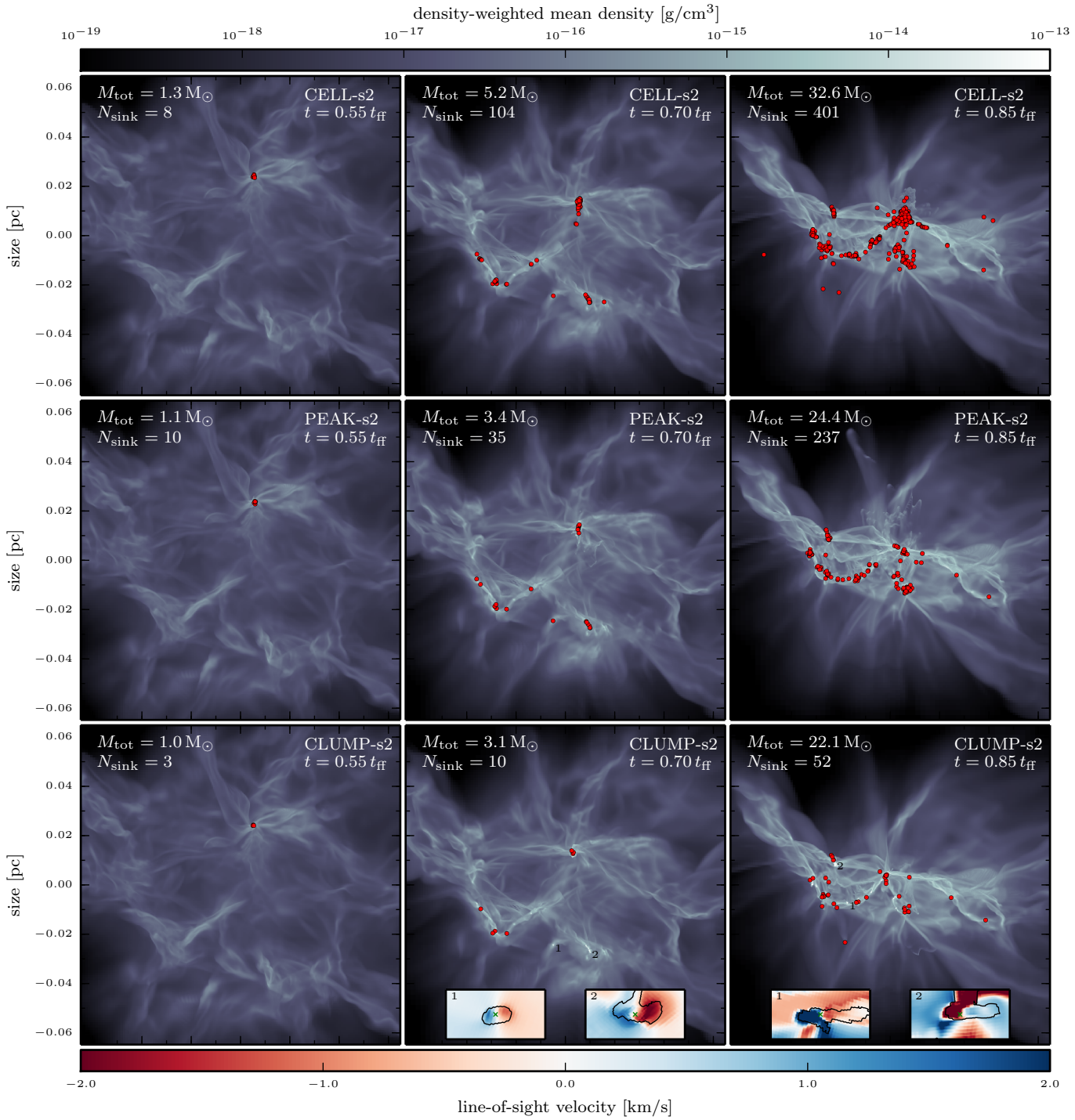


Figure 3.11: The same as Figure 3.10 but for the s2 setup.

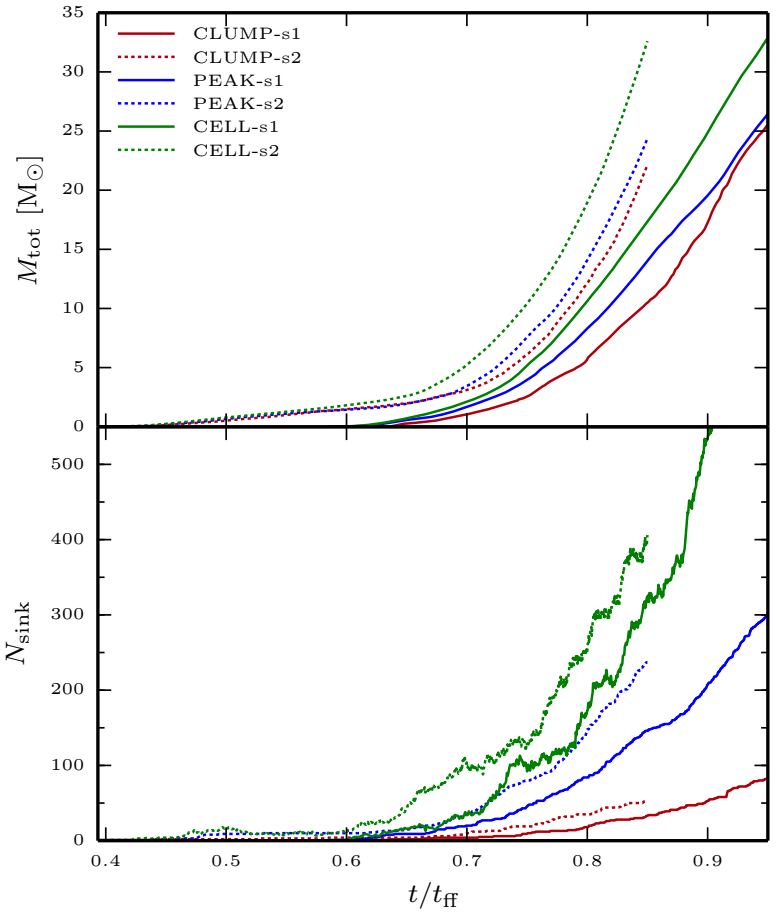


Figure 3.12: Temporal evolution of the number of sinks $N_{\rm sink}$ and the total mass in sinks $M_{\rm tot}$ for the six runs.

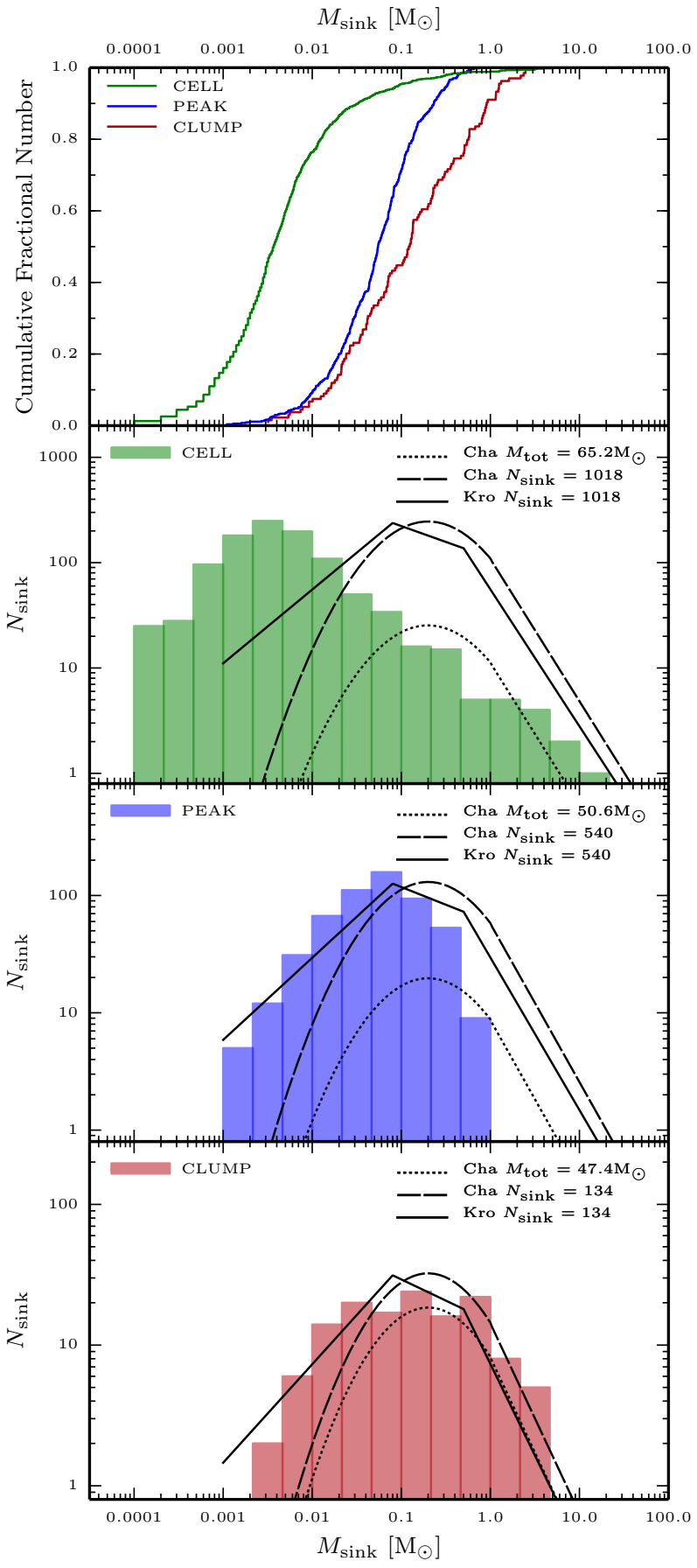

$M_{\rm sink}$ [M$_\odot$]
0.0001
0.001
0.01
0.1
1.0
10.0
100.0
Cumulative Fractional Number
CELL
PEAK
CLUMP
$N_{\rm sink}$
Cha $M_{\rm tot}$ = 65.2M$_\odot$
Cha $N_{\rm sink}$ = 1018
Kro $N_{\rm sink}$ = 1018
Cha $M_{\rm tot}$ = 50.6M$_\odot$
Cha $N_{\rm sink}$ = 540
Kro $N_{\rm sink}$ = 540
Cha $M_{\rm tot}$ = 47.4M$_\odot$
Cha $N_{\rm sink}$ = 134
Kro $N_{\rm sink}$ = 134

Figure 3.13

Figure 3.13: Joint (resulting from the s1- and s2-setup) sink mass distributions for the different sink formation criteria. The top panel shows the cumulative fractional number for each of the three mass distributions in one plot. Below we plot the individual mass histograms together with the Chabrier (2005) IMF normalized to the total mass in sinks and to the number of sinks respectively as well as the Kroupa (2001) IMF normalized to the number of sinks. A Kolmogorov-Smirnov test confirms that all sink mass distributions are different.

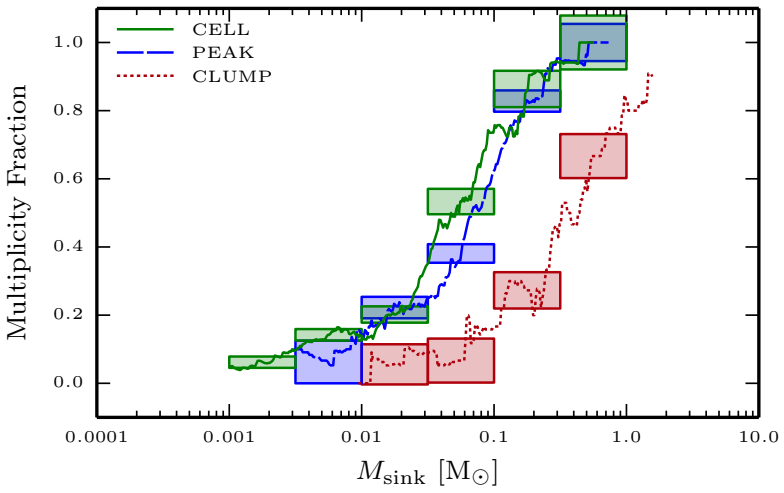


Figure 3.14: Multiplicity fraction as a function of primary mass. The width of the boxes corresponds to the primary mass bins and the height gives the $\pm 1\sigma$ range. Only those mass bins with at least 10 objects are considered. The continuous lines show the corresponding boxcar-averages.

Table 3.3: Statistical properties of the sinks formed collapsing turbulent gas using different sink formation algorithms. We show the total number of sinks, the total mass in sinks, the average mass, the median mass and the mass of the heaviest sink at the end of each run. The last column contains the width of the sink mass distribution in log-space.

	IC	$N_{\rm tot}$	$M_{\rm tot}\,[{\rm M}_\odot]$	$\overline{M}\,[{\rm M}_\odot]$	$\tilde{M}\,[{\rm M}_\odot]$	$M_{\rm max}\,[{\rm M}_\odot]$	$\sigma\Big(\log_{10}(M/{\rm M}_\odot)\Big)$
CELL	s1	620	32.7	0.053	0.0042	9.77	0.69
CELL	s2	398	32.4	0.082	0.0030	11.67	0.76
PEAK	s1	303	26.3	0.087	0.053	0.78	0.51
PEAK	s2	237	24.3	0.10	0.060	0.75	0.50
CLUMP	s1	82	25.4	0.31	0.080	2.65	0.71
CLUMP	s2	52	22.0	0.42	0.18	2.49	0.63

While all sink creation methods agree in the fact that the s1-run forms ≈ 1.5 times as many sinks as the s2-run, the number of sinks formed and therefore the average sink mass strongly differ. Considering the results of both setups together, the CLUMP algorithm reduces the number of sinks by 87 per cent when compared to the CELL algorithm and by 75 per cent when compared to the PEAK strategy. In Figure 3.13 we analyze the joint sink mass functions from both setups for each sink formation algorithm. In the top panel we display the cumulative mass functions. Kolmogorov-Smirnov tests yield p-values below 10^{-8} for each pair of mass functions which means that the underlying distributions are different. The absence of further checks for sink formation leads to very high number of low mass sinks produced by the CELL algorithm. Furthermore, the aggressive merging strategy increases the accretion rate of already heavy objects which results in a flat high mass tail and one object with a mass $\sim 10\,\mathrm{M}_\odot$ formed in each run. The PEAK and the CLUMP runs produce similarly shaped mass distributions which resemble the observed IMF (Chabrier, 2005; Kroupa, 2001). The PEAK distribution is shifted to lower masses and has a somewhat steeper drop-off at high masses compared to the CLUMP distribution. We find a good qualitative agreement between the PEAK results and the top-hat results obtained by Girichidis et al. (2011) for equivalent setups and a very similar sink formation algorithm. The surprisingly good agreement between our new sink mass function and the observed IMF (see bottom panel of Figure 3.13) must be seen (at least partially) as a coincidence. The rather low sink formation density threshold of $2.46 \times 10^{-14}\,\mathrm{g/cm^3}$ and the warm temperature of 20 K that we have adopted in the numerical experiment both lead to a high minimum Jeans mass which increases the characteristic mass of the produced sinks. These rather arbitrary choices are unavoidable, because we do not model the effect of radiative feedback in setting up the characteristic star particle mass (e.g., Krumholz et al., 2012).

In Figure 3.14 we compare the multiplicity fractions at the end of the simulations. As for the mass functions, we add the results from the s1 and the corresponding s2-run. We adopt the following definition (Hubber & Whitworth, 2005) of the multiplicity fraction

$$mf = \frac{B + T + Q}{S + B + T + Q}, \tag{3.45}$$

where S is the number of single objects and B,T,Q are the number of binary, triple and quadruple *systems* respectively that have a primary mass in a given range. We follow the algorithm described by Bate (2009) to group the sinks into gravitationally bound systems. Despite the relatively high uncertainty in our results due to the low number of objects per mass bin, one can safely conclude that for the chosen setup a sink with a mass in the range $[0.1\,\mathrm{M}_\odot, 1.0\,\mathrm{M}_\odot]$ has a significantly lower probability to have companions when we use our new sink formation algorithm. We interpret this effect as being due to the correct treatment of tidal forces in our virial check, which hinders the formation of new sinks close to pre-existing ones.

3.7.2.1. Sink merging comparison

Table 3.4: Results of the sink merging comparison runs. N_+ stands for the number of sink formation events, N_- for the number of mergers. Furthermore we display the total number of sinks, the total mass in sinks, the average mass, the median massn the mass of the heaviest sink and the width of the sink mass distribution in log-space at the end of each run.

	IC	N_+	N_-	N_{tot}	$M_{\mathrm{tot}}\,[\mathrm{M}_\odot]$	$\overline{M}\,[\mathrm{M}_\odot]$	$\tilde{M}\,[\mathrm{M}_\odot]$	$M_{\mathrm{max}}\,[\mathrm{M}_\odot]$	$\sigma\Big(\log_{10}(M/\mathrm{M}_\odot)\Big)$
nomerge	s1	82	0	82	25.4	0.31	0.08	2.65	0.71
nomerge	s2	52	0	52	25.2	0.42	0.16	2.49	0.63
allmerge	s1	102	52	50	22.8	0.46	0.07	8.90	0.77
allmerge	s2	51	23	28	17.0	0.61	0.06	7.30	1.00
$t_{\mathrm{merge}} = 5000\,\mathrm{yr}$	s1	106	54	52	24.9	0.49	0.07	5.11	0.86
$t_{\mathrm{merge}} = 5000\,\mathrm{yr}$	s2	71	37	34	20.1	0.59	0.11	3.72	0.83
$t_{\mathrm{merge}} = 1000\,\mathrm{yr}$	s1	106	37	69	25.0	0.36	0.10	3.10	0.78
$t_{\mathrm{merge}} = 1000\,\mathrm{yr}$	s2	69	30	39	22.6	0.58	0.16	5.39	0.68
$t_{\mathrm{merge}} = 500\,\mathrm{yr}$	s1	69	9	60	23.0	0.38	0.10	2.14	0.72
$t_{\mathrm{merge}} = 500\,\mathrm{yr}$	s2	46	3	43	22.4	0.52	0.20	3.73	0.64

We use the same turbulent core to test the influence of sink merging onto sink formation and accretion. In Section 3.4.1 we introduced the concept of a merging timescale allowing only young sinks to merge. The same timescale is used as maximum timescale (or a minimum speed) at which the gas must contract in order to form a sink. The physical motivation for this merging of young sinks is the finite lifetime of the first Larson core (Larson, 1969; Masunaga et al., 1998) of $\sim 1000\,\mathrm{yr}$ during which the sink represents an "fluffy" uncollapsed object. We thus compare the results from the previous section where sink merging is turned off to runs where three merging time scales $t_{\mathrm{merge}} = 500\,\mathrm{yr}$, $1000\,\mathrm{yr}$, $5000\,\mathrm{yr}$ and a case where we allow sinks to merge during their entire lifetime corresponding to a infinite merging timescale. Some statistical properties of the sinks formed in each run are listed in the Table 3.4.

In Figure 3.15 we plot the temporal evolution of the number of sinks and the mass in sinks together with the cumulative sink mass distribution and the multiplicity fraction as a function of primary mass. Comparing the two limiting cases (no merging, infinite merging lifetime) we find that sink merging reduces the number of sinks by ≈ 40 per cent. Furthermore we see a strong increase in the mass of the heaviest sink together with slight decrease of the median sink mass, resulting in wider mass distribution. However, the data generated in this test is rather scarce. A Kolmogorov-Smirnov test returns a p-value of 7 per cent when comparing the joint (s1 together with s2 run) distributions resulting from the "nomerge" and the "allmerge" runs. Yet the observed trend fits well with our results for the CELL algorithm in the previous section which merges sinks in a FOF-fashion and produces a wider mass distribution and a couple of very high mass objects too. In the s1-run merging increases the number of sink formation events which suggests that the region close to the site of a merger will often create another sink. However, this seems to be very setup dependent as the s2 run shows a different picture. Here, merging decreases the total mass in sinks through the early formation of a very heavy object that prevents sinks from being formed in its surrounding.

As one expects, sink merging decreases the number of sinks in multiple systems. The bottom-right panel of Figure 3.15 shows a reduction of the multiplicity fraction by ≈ 50 per cent for primary masses in the range $[0.1\,\mathrm{M}_\odot, 1.0\,\mathrm{M}_\odot]$ when sinks are merged.

We now take a look at the three cases where we used a finite merging timescale. A merging timescale of 5000 yr gives results similar to the "allmerge" case as most of the sink formation is happening within $0.2\,t_{\mathrm{ff}} \approx 10\,\mathrm{kyr}$. The reduction of the heaviest sink masses shows that the very high mass objects produced by the "allmerge" runs form through late time mergers. The shorter merging timescales lead to results which are more similar to the "nomerge" runs, following a trend for lower maximum mass, slightly higher mean mass and narrower mass distribution for shorter merging timescales. For $t_{\mathrm{merge}} = 500\,\mathrm{yr}$ the usage of the merging timescale to define a minimum contraction rate for sink formation starts to kick in, leading to less formation events than in the case without merging.

We now change our focus to accretion onto sink particles. Accretion can influence formation and merging of sink particles by producing new peaks in the gas density field which might trigger sink formation. It is therefore desirable to have an accretion scheme which produces a smooth transition of the flow variables at the sink accretion boundary.

3.7.3. Spherical bondi accretion

There are two setups which are frequently used to test the accretion of sink particles from spherically symmetric gas configurations: The collapse of a singular isothermal sphere

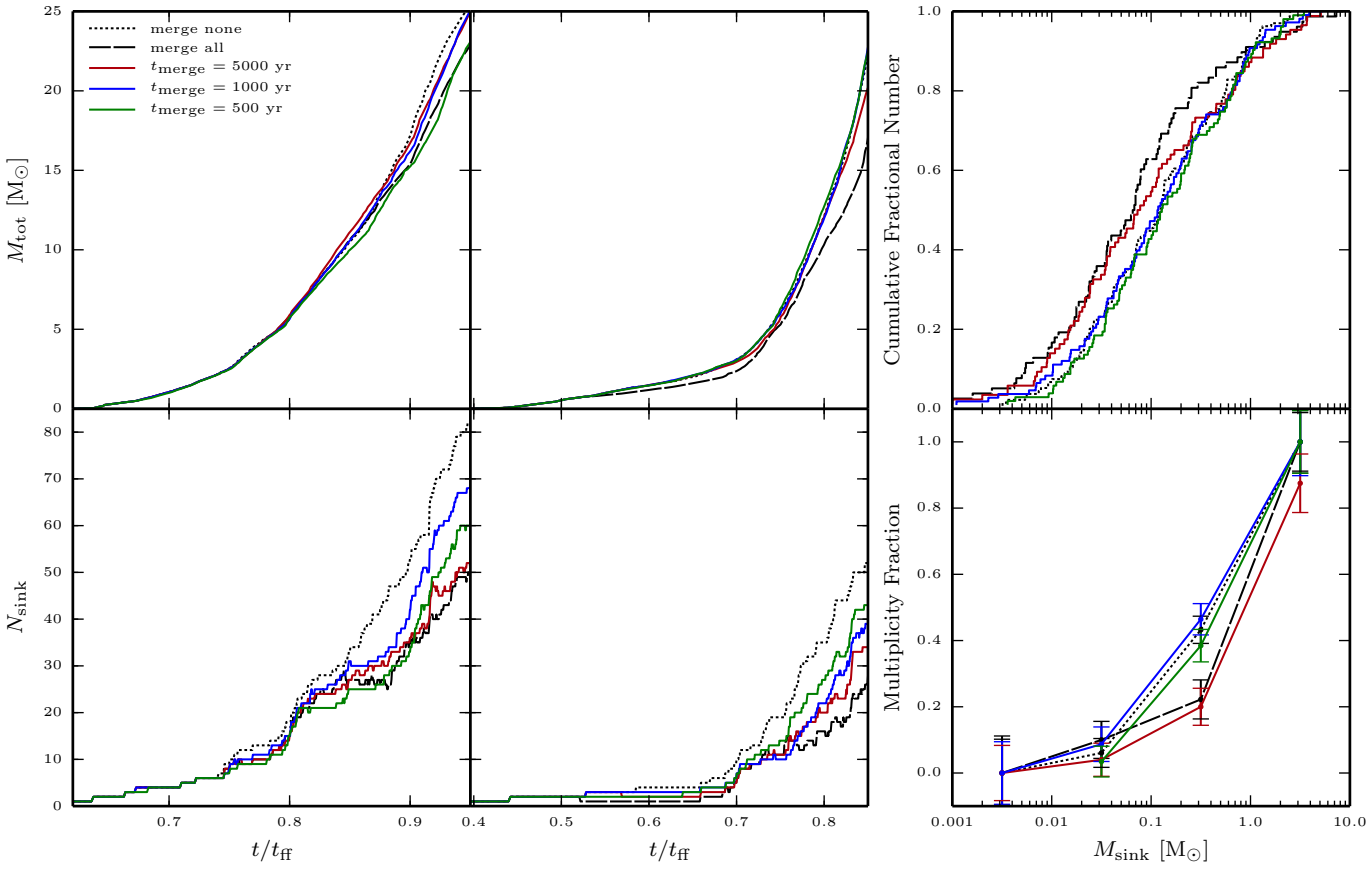


Figure 3.15: Sink merging comparison. The two panels on the left show the number of sinks and the total mass in sinks as a function of time for the s1 setup. The panels in the middle display the corresponding plots for the s2 setup. The upper right panel shows the cumulative sink mass distributions and in the lower right panel we display the multiplicity fraction as a function of primary mass, where each datapoint covers one order of magnitude in primary masses.

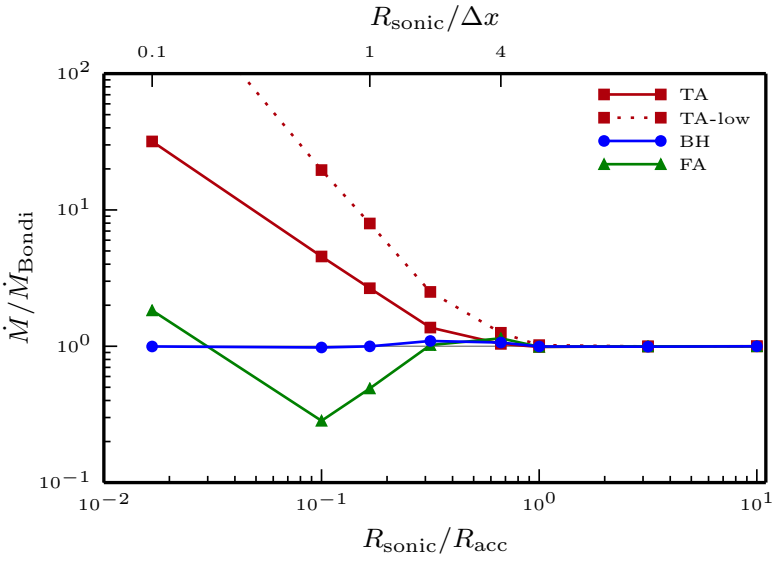


Figure 3.16: Normalized accretion rates as a function of the sonic radius when simulating the Bondi problem with the different accretion schemes.

first studied by Shu (1977) and Bondi accretion (Bondi, 1952). While the collapse of the isothermal sphere is usually well modeled by the codes using sink particles (Federrath et al., 2010; Gong & Ostriker, 2013; Krumholz et al., 2004) the Bondi accretion test is a harder challenge as soon as the infall velocity of the gas onto the sink particle is smaller than or of the same order as the sound speed (Hubber et al., 2013; Krumholz et al., 2004). We therefore follow the latter two authors and test how well the different accretion methods recover Bondi's transsonic isothermal solution for the accretion of a star at rest relative to the surrounding gas,

$$\dot{M}_{\rm Bondi} = \frac{\pi \exp(3/2) G^2 M_\star^2 \rho_\infty}{c_\infty^3}, \tag{3.46}$$

where all the quantities have the same meaning as in Section 3.6.2. We place a spherically symmetric gas ball in a simulation box with an effective resolution of 512^3 cells. The radius of the ball is 128 cells or 25 per cent of the box. The initial density and velocity field inside the ball are chosen according to the numerical solution of the Bernoulli equation. Outside, the density obtained by the numerical solution is multiplied by 10^{-4} and the cells are de-refined by two levels. A sink particle is placed at the center of the box. The accretion radius $R_{\rm acc}$ is set to 6 cells and the gravitational softening radius $R_{\rm soft}$ of the sink is 3 cells[14]. The sink threshold density is picked according to the numerical solution at the location of the sink boundary. We employ the PM method for computing sink-gas interactions. Using direct force summation instead yields almost identical results. In this test, the total gas mass is negligible compared to the sink mass justifying the assumption of a constant gravitational field.

In the transsonic Bondi solution the sonic radius

$$R_{\rm sonic} = \frac{GM_\star}{2c^2} \tag{3.47}$$

[14] As Krumholz et al. (2004) we find the results to be more accurate when the force at the sink accretion boundary is given by the unsoftened value.

separates regions of supersonic gas velocities (inside $R_{\rm sonic}$) from regions of subsonic flows (outside $R_{\rm sonic}$). Varying the sink mass therefore sets the ratio $R_{\rm sonic}/R_{\rm acc}$ which determines whether the inflow through the sink boundary is subsonic or supersonic. The simulations are stopped at $t_{\rm end} = 4R_{\rm acc}/c$ which is after the accretion rates have reached constant values but before the rarefaction wave from the boundary enters the scene as $4R_{\rm acc} \ll R_{\rm sphere}$. We compare the accretion rates at the end of the simulations to the analytical Bondi rates given by Equation 3.46 and plot it in Figure 3.16 against the ratio $R_{\rm sonic}/R_{\rm acc}$ for the different accretion schemes.

In the supersonic regime $R_{\rm sonic} > R_{\rm acc}$ all the simulated accretion rates differ by less than 1 per cent from the Bondi rate and even when the sonic Radius and the sink radius are the same, all accretion rates are within 5 per cent from the analytic value. Modifications of the density fields within the sink accretion radius are "hidden" from the rest of the simulation domain since no wave can propagate outward from the accretion zone. In the subsonic regime $R_{\rm sonic} < R_{\rm acc}$ the simulated accretion rates differ by many orders of magnitude. Altering the density inside the sink accretion zone now does affect the accretion rate. For instance, an overestimation of the *initial* accretion rate can cause a sharp drop in the density at the boundary of the accretion zone which triggers an outward traveling rarefaction wave and therefore leads to a *permanent* overestimation of the accretion rate.

Not surprisingly the BH accretion scheme performs best at what it was designed for - solving the Bondi problem. For the BH case we find the biggest deviation from the analytic value when the sonic radius is in between the sink accretion radius and the grid spacing. At $R_{\rm sonic}/R_{\rm acc} = 0.31$ we overestimate the accretion rate by 9 per cent. For the regime where $R_{\rm sonic} \geq R_{\rm acc}$ or $R_{\rm sonic} \leq \Delta x$ the errors are smaller than one per cent. This is similar to Krumholz et al. (2004) who find a deviation of ≈ 25 per cent from the analytic value when the accretion radius is of the same order as the Bondi radius. In the TA case, the accretion rates are very sensitive to the chosen threshold as soon as $R_{\rm sonic} < R_{\rm acc}$. Even though we artificially set the sink threshold to the analytic value at the location of the sink accretion boundary, the accretion rate is overestimated. The results are obviously worse for the TA-low case as reducing the density inside the accretion zone reduces the back pressure on the flow outside the accretion zone. On the other side, increasing the threshold by one order of magnitude stops accretion completely in that regime (not plotted). The FA scheme seems to perform acceptably in this test on the first sight, yet it suffers from a different problem: For the runs where $R_{\rm sonic}/R_{\rm acc} \leq 0.31$ the accretion rates do not converge during the course of the simulation. Instead of the final value we therefore plot the *average* accretion rates for those data points. The FA scheme lets the sink accrete exactly at the Bondi rate at the beginning of the simulation since the mass flux into the accretion zone is correctly set by the initial conditions. When running the simulation long enough, the accretion rate starts to oscillate with a growing amplitude, temporarily even dropping to zero. We interpret this behavior in the following sense: Stability analysis of the Bondi problem (e.g., Stellingwerf & Buff, 1978) have shown that only the transsonic solution to the problem is stable. When the resolution is very limited ($R_{\rm sonic} \lesssim \Delta x$) there is no region where the flow is supersonic and the solution to the problem becomes indistinguishable from solutions without a supersonic region and therefore unstable. The reason that we do not see this instability for the BH and the TA case is that for those schemes, the accretion rates are effectively monotonic functions of the density inside the accretion zone what stabilizes those solutions.

Table 3.5: Simulation parameters for the disk accretion tests.

Sphere radius	R	$= 2000\,\mathrm{au}$
Total gas mass	M	$= 2\,\mathrm{M}_\odot$
Density profile	$\rho(r)$	$= \frac{\rho_0}{(r/r_0)^2+1}$
	ρ_0	$= 4.7 \times 10^{-13}\,\mathrm{g/cm^3}$
	r_0	$= 10\,\mathrm{au}$
Isothermal sound speed	c_s	$= 1.88 \times 10^4\,\mathrm{cm/s}$
$E_\mathrm{therm}/E_\mathrm{grav}$	α	$= 0.06$
Angular velocity	Ω	$= 5.45 \times 10^{-12}\,\mathrm{s^{-1}}$
$E_\mathrm{rot}/E_\mathrm{grav}$	β	$= 0.33$
Box size	L_box	$= 32\,000\,\mathrm{au}$
Cell size at levelmax	Δx_min	$= 7.8\,\mathrm{au}$
Sink accretion radius	R_acc	$= 4\Delta x_\mathrm{min}$
Sink softening radius	R_soft	$= 2\Delta x_\mathrm{min}$

3.7.4. Disk accretion tests

Sink particles in simulations of self gravitating turbulent gas accrete most of their mass from the disks that form around them. Since there is no appropriate toy model with analytical solution for this mode of accretion, we have to compare results obtained by using different accretion schemes to each other without knowing the "true" solution. We do this by studying the collapse of a rotating gas sphere which triggers the formation of a sink particle surrounded by an accretion disk. The parameters describing the initial setup together with some simulation parameters are listed in Table 3.5. Since we do not use radiative feedback in these calculations, we use a piecewise polytropic EOS

$$P = \begin{cases} {c_s}^2\rho & \text{if} \quad \rho \le 1.0 \times 10^{-16}\,\mathrm{g/cm^3}, \\ \kappa\rho^{1.4} & \text{if} \quad \rho \ge 1.0 \times 10^{-16}\,\mathrm{g/cm^3}, \end{cases} \tag{3.48}$$

to heat the dense gas and prevent the disk from fragmenting into multiple sinks. κ is chosen such that P is a continuous function of ρ. In this test we use direct force summation for computing sink-gas interactions[15].

At $t = 1.61\,\mathrm{kyr}$ a sink forms at the center of the sphere. Very quickly after its formation a marginally unstable disk starts to develop around the sink. At $t = 5\,\mathrm{kyr}$ the diameter of the disk has reached $\approx 200\,\mathrm{au}$. By this time, the accretion rate has dropped to several $10^{-5}\,\mathrm{M_\odot yr^{-1}}$. We let the sink accrete from that disk until we stop the simulation at $t_\mathrm{end} = 100\,\mathrm{kyr}$.

The sink masses and accretion rates as a function of time are plotted in Figure 3.17 together with the disk density- and radial velocity profiles at $t = 50\,\mathrm{kyr}$. The simulated accretion rates differ strongly right after the sink formation. During this phase the disk

[15]Comparison runs using the PM scheme show similar behavior for the first $\approx 5\,\mathrm{kyr}$, but tend to loose symmetry quickly once the sink is growing massive and therefore dominating the gravitational potential. This causes the sink to leave the center of the disk which considerably changes the results.

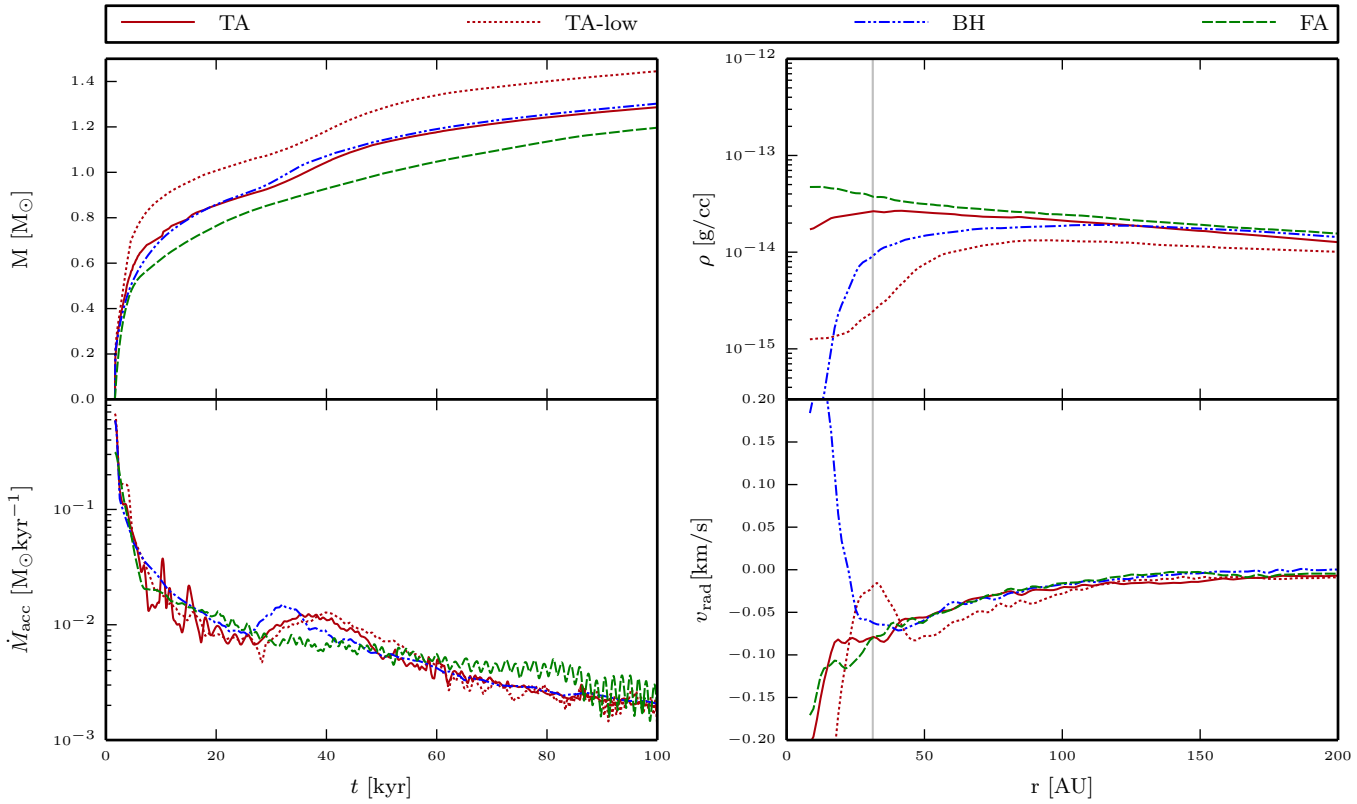


Figure 3.17: Accretion from a disk onto the central sink particle using different accretion schemes. The two panels on the lefthand side display mass and accretion rate of the central sink particle as a function of time. The panels on the right hand side show the corresponding disk profiles at $t = 50$ kyr. The disk profiles are computed in a mass weighted fashion considering all cells within ± 39 au from the mid plane which corresponds to roughly one scale height in each direction. The vertical gray line at ≈ 30 au in the profiles indicates the sink accretion radius.

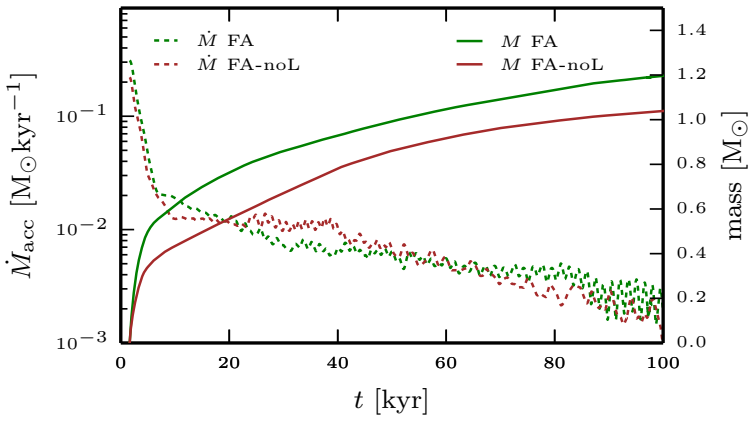


Figure 3.18: Mass and accretion rate onto the central sink particle when we use no-L accretion compared to the standard flux accretion case.

can efficiently dispose of angular momentum by accreting it into the sink and pressure gradients still play an important role in controlling accretion. A high temporary accretion rate can therefore lead to a high permanent accretion rate. As soon as the disk surrounding the sink is a few times the size of the accretion zone, the accretion rates tend to converge for the different accretion recipes, thereby conserving the differences in their masses that they have obtained in the first $\approx 10\,$kyr. In this phase we see a self-regulating effect even in the absence of radiative feedback from the sink particle: Lower accretion rates lead to higher densities in the disk which promotes the development of spiral arms. These spiral arms facilitate the re-distribution of angular momentum and therefore increase the accretion rate. The upper right panel in Figure 3.17 shows that the density in the center of the disk does depend on the chosen accretion scheme. Because of the subsonic radial velocities (lower right panel) these changes are not restricted to the sink accretion zone but affect the density profile out to several accretion radii. The way accretion is controlled in the BH and TA scheme favors accretion from cells very close to the sink. Together with the centrally peaked accretion kernel used in the BH run, this leads to drop in the density by more than two orders of magnitude. In the TA runs, the depth of this central hole is limited by the accretion threshold. No such hole is produced by the FA run which shows the smoothest transition of the flow into the accretion zone.

3.7.4.1. Accretion of angular momentum

The above tests have been performed without conserving angular momentum in the gas when accreting onto the sink. We briefly study the effect of what we call no-L accretion (see Section 3.6.4) in combination with the FA accretion method using the above setup. Hubber et al. (2013) find that angular momentum feedback from the sink back to the SPH particles considerably lowers the accretion rate during the first $\approx 1\,$kyr when following the collapse of a rotating Bonnor-Ebert sphere. We plot our results in Figure 3.18. Even though AMR as a fundamental difference to SPH does not conserve angular momentum, our results agree well with those found by the authors mentioned above for the early evolution of the sink. During the first 10 kyr after the formation of the sink, the average accretion rate is reduced by 30 per cent when no-L accretion is used. As soon as the disk is big compared to the accretion radius, the amount of angular momentum that can be advected into the

sink particle is small compared to the angular momentum in the disk and the accretion rates for the two runs are very similar. Although not huge, this difference in the early accretion rate might still be enough to affect the probability of a core to fragment into a multiple system rather than a single object.

3.8. Conclusions

We presented a new method for sink particle creation and its implementation in the AMR code RAMSES. The new method uses a clump finder to identify well-defined density peaks as possible locations for sink formation. We discussed previously introduced tests that are used to examine the gas surrounding a density peak for gravitational collapse and suggested a new criterion based on a virial equilibrium type analysis that fully respects the tidal fields caused by the surrounding mass distribution. We argue that this is more physically motivated than existing criteria. We compared the new method to the most frequently used sink creation recipes in simulations of gas undergoing gravitational collapse. Overall, we found our new algorithm to be more restrictive and it triggers less sink formation than other techniques. We showed that none of the sink particle implementations can prevent artificial fragmentation of a filament that formed in an isothermal Boss & Bodenheimer test. However, our new method is less susceptible to the formation of sinks from those artificial fragments. We simulated the collapse and fragmentation of a small isothermal molecular cloud and found that the number of sink particles formed varies by up to a factor of eight depending on the sink formation algorithm used. The median values of the obtained sink masses differ by up to a factor of 60 and the most massive sinks produced in each run vary by more than one order of magnitude. In the same test, our new algorithm gives rise to a lower probability for sinks to be part of a multiple system than the comparison runs. We do not repeat the analysis for non-isothermal gas, but performing a Boss & Bodenheimer test using a polytropic EOS suggests that the statistical properties of the sinks formed in non-isothermal turbulent gas will depend on the sink creation routine as well. We therefore conclude that the usage of (different) sink algorithms limits the comparability of results in star cluster formation simulations. Furthermore, great care must be applied when interpreting results that are obtained from such calculations.

We discussed merging of sink particles and describe an intermediate scenario that allows sinks to merge during a certain time-span. We tested sink merging on the turbulent cloud setup. In combination with our new sink creation routine we found that sink merging reduced the number of sinks up to a factor of two when sinks are allowed to merge during their entire lifetime. The information obtained in this test does not allow us to make definite statements about the influence of merging on the sink mass distribution and multiplicity function, but we observed a trend towards a small increase in the width of the mass distribution and a decrease of the multiplicity fraction when merging is allowed. More significantly, sink merging does increase the mass of the most massive sink produced in a calculation. These effects of sink merging can be expected to be even larger when a less restrictive sink formation algorithm is used.

We implemented and compared two schemes for computing sink-sink and sink-gas forces: a PM method and a direct force summation approach. The PM scheme produces surprisingly stable orbits when we let two sinks orbit each other on elliptical trajectories as long as all AMR level boundaries are sufficiently far away. When a sink particle which dominates the local gravitational potential gets close to a level boundary, spurious forces arise that can artificially influence the results. On the other hand, we obtain a speed gain

of the order of $\sim (1 + \frac{n_{\rm sink}}{100})$ for a typical setup when using the PM method. A possible way to improve on this situation is to include only the most massive sinks in the direct force summation and to treat the lighter ones using the PM method.

We have implemented different methods to perform accretion onto sink particles and tested these on two different simulations. The case of spherical Bondi accretion is well modeled by all methods as long as the infall velocity through the sink accretion radius is supersonic. When the accretion is subsonic, only the usage of the Bondi formula for computing the accretion rate will give a correct and stable result in the long run. When accreting from a disk, all accretion schemes yield similar results as soon as the disk radius is larger than a few accretion radii. However, the density and velocity profile of the region close to the sink can be affected considerably. We find that flux-accretion produces the smoothest profiles without any violent changes of the hydrodynamic variables at the sink boundary. Furthermore, flux accretion naturally adapts the accretion rate in the case of a disk with no need for evaluating specific energies on a cell-by-cell basis. These properties lead us to adopt flux accretion as our standard accretion scheme for sink particles in RAMSES. However, for situations where the the sonic radius of the sink is smaller than the sink accretion radius (as it can be the case for a sink inside an object undergoing Kelvin-Helmholtz contraction) we recommend switching to the Bondi rate. This can be achieved automatically by the simulation code. We implemented so called "no-L accretion" where we leave the angular momentum in the remaining gas which is not accreted. Comparing this to the case where sinks act as sinks for the angular momentum as well yields a considerable reduction of the obtained accretion rate from the disk in the early stage after the sink formation. Once a large disk has formed around the sink, the difference in the accretion rates is negligible.

3.9. Acknowledgements

The computations leading to this publication have been performed at on the zBox4 and Schroedinger Supercomputers at the University of Zurich and at the Swiss Supercomputing Centre CSCS in Lugano. This work has been supported by the Swiss National Science Foundation SNF under the project "Computational Astrophysics" and the PASC co-design project "Particles and Fields'.

3.10. Appendix A: elliptical orbit test

The next test concerns the ability of the algorithm to produce accurate sink particle trajectories. We let two equal mass sink particles orbit their common center of mass on elliptic trajectories in the absence of gas. The initial separation of the two sinks is 24 cells and the initial velocities are chosen such that the minimum separation of the two sinks is 6 cells if they move on their analytically predicted orbits. We use this setup to compare the PM force calculation for sink particles with the direct force summation. We furthermore distinguish the PM case into a run where the grid is fully refined to level 7 (PM case) and another setup where the cells further than $6\Delta x$ from the sink particle are allowed to de-refine to level 6 (PM-AMR case). When AMR is activated we enforce single time stepping, meaning that the coarser level is updated using the same time step as the finer level. The Plummer softening length is set to $2\Delta x$ in the direct force summation run and the radius of the particle "swarm" is set to $3\Delta x$ for the PM cases. We measure total energy and angular momentum in the system during the first 20 orbits and plot the results

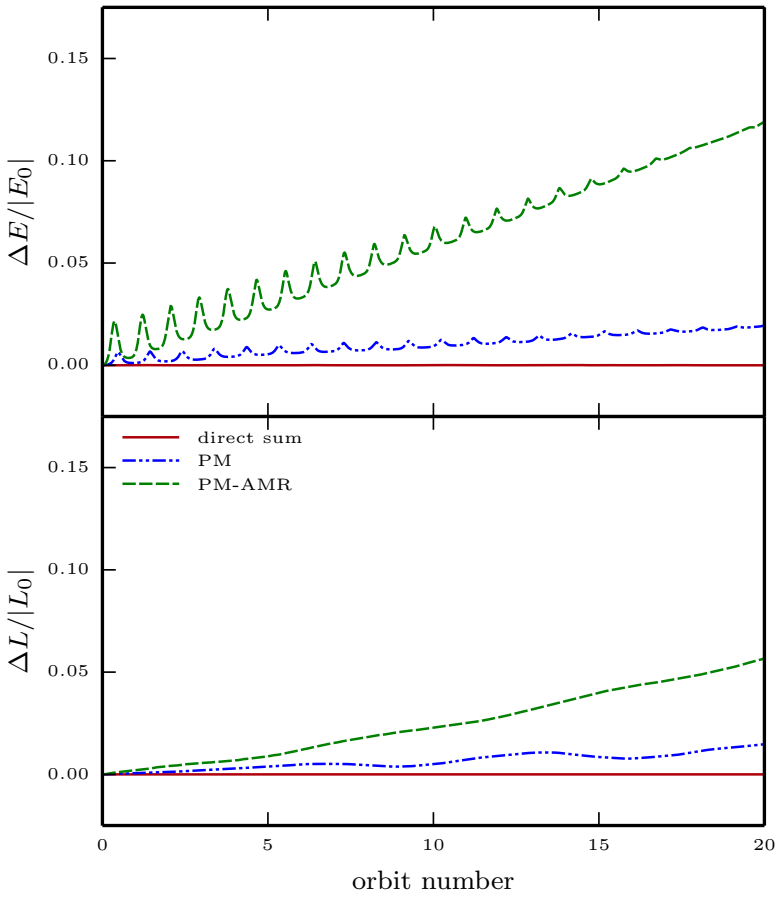


Figure 3.19: Conservation of energy and angular momentum in the sink binary system obtained using different force calculation methods. The plotted quantities are smoothened over the analytically computed orbital time.

in Figure 3.19.
The direct force summation delivers excellent results in this test. For this setup the time step is controlled by the free fall time criterion (see Equation 3.30) which leads to a constant time step as long as the sink masses are constant. For a constant time step, the particle integration scheme is equivalent to a leapfrog integrator and therefore obtains its symplectic property. Angular momentum and energy are thus conserved to machine precision. We see a considerable precession of the perihelion by $\approx -7.7°$ per orbit which is caused by the deviation from the $1/r$-potential induced by the softening. The results for the PM scheme in the absence of AMR are surprisingly good. The picture changes dramatically when AMR is turned on. The main source of problems for the PM scheme are the level boundaries. The poisson solver in RAMSES uses a "one-way interface" scheme (Guillet & Teyssier, 2011) which means that the coarse level potential is used to set boundary conditions for the refined regions. This is problematic since a poorly resolved mass distribution (as it is the case for the sink on coarse levels) leads to large errors in the potential.

3.11. Appendix B: sink integration speed

The previous section shows the superior accuracy of the direct force summation over the PM approach. However, there is still good reason to use the PM scheme in order to accelerate calculations involving a "large" number of sinks. In this short subsection we estimate the speed gain that can be expected when using the PM method. We consider an initially homogeneous, slightly turbulent gas sphere at level 8. We then randomly place $n_{\rm sink}$ equally massive sink particles inside the sphere. The total gas mass is identical to the total sink mass. We then let the code refine around the sink particles up to level 18. As soon as the refinements are done and the usual load balancing has been performed, we measure the time needed to perform ten time steps. Wo obtain the following speedup when using the PM method compared to direct force summation:

$n_{\rm sink}$	10	100	1000	10000
PM-acc	1.1	1.7	6.3	46.3

In all cases, the number of gas cells hosted by each MPI process is much larger than the total number of sink particles. When computing the forces directly, it is therefore the $n_{\rm sink}$ loops over all its cells that each MPI process has to perform which are dominating the extra execution time. We can therefore estimate that a usual hydro and gravity time step by RAMSES roughly takes takes execution time of 100 loops over all cells. So if $n_{\rm sink} \gg 100$ the total execution time is dominated by the direct force summation and we recommend switching to the PM method. Here our results differ from what Federrath et al. (2010) find for their implementation into the FLASH code. When computing the direct sum of all sink-gas interactions, their total execution time is not significantly increased for ~ 1000 sinks.

4

The IMF from Simulations of Star Cluster Formation

In this chapter I focus on attempts to model the IMF through simulations of star cluster formation in molecular clouds. I start with a short discussion of the main challenges in this field and some contemporary trends. I describe a computationally inexpensive treatment of infrared feedback from accreting protostars and validate this by comparing to a calculation that self-consistently solves the equations of radiation hydrodynamics. We then prepare the grounds for future simulations by comparing two possible criteria to control the creation of new AMR refinements during the collapse of the cloud. Finally, I report on a suite of simulations that model a collapsing 1000 $M_\odot$ molecular cloud and analyze the emerging IMF.

As already stated in the introductory part of this thesis, star formation is a *multi-physics* and *multi-scale* process. This makes it not only a problem which is hard to tackle by analytic means, it also turns simulating the process of star cluster formation into a very challenging task. Large efforts have been made to advance on both fronts: the application of the AMR and the SPH technique has alleviated the *multi-scale* problem. On the multi-physics side, astrophysical simulation tools like RAMSES have been equipped with additional code modules to treat magnetic fields, radiation or the chemical evolution of the gas accurately. However, modeling all relevant processes over the entire range of scales, from parsecs down to several solar radii, from the masses of molecular clouds to those of brown dwarfs - possibly during several million years - is computationally still not feasible. It is thus unavoidable to reduce the size of the problem in some aspects. The introduction of sink particles described in the previous chapter achieves this by limiting the smallest length scale in the problem to several AU, but further restrictions are necessary to obtain a solvable problem size. Decreasing the resolution by lowering the density for sink formation makes it impossible to resolve brown dwarfs and the resulting sink mass functions are thus only sampling a part of the IMF. Another strategy is to reduce the size and the mass of the initial molecular cloud. But this of course also reduces the total mass present in stars when the simulation is stopped. As a result, one is left with results of very little statistical power, especially on the high mass end of the IMF. For example, a $1000\,\mathrm{M}_\odot$ that is stopped once 10% of the total gas mass have turned into stars, should in average contain roughly one star with a mass $10\,\mathrm{M}_\odot$ or more. As a third option one can try to simulate large clouds at high resolution, while neglecting certain physical aspects completely, or trying to approximate them in a computationally inexpensive way. Results obtained in such a way have to be interpreted with a lot of care as the observed properties can always be contaminated by shortcomings of the model. In summary, we are facing a situation where a trade-off between physical accuracy and speed - and hence statistical significance of obtainable results - has to be made. In this section I describe our attempts to handle this situation, aiming for big, high-resolution simulations delivering statistically meaningful samples of the IMF.

Previous studies aiming to shed light on the question of the origin and the universality of the IMF represent different approaches to the described challenge. As suggested by Hennebelle & Chabrier (2008) theory, the core mass function (CMF) has a key role in shaping the IMF. Some authors have therefore used the clump as the natural entity to divide star formation in giant molecular clouds into two separate problems, each of them with a smaller dynamic range. In all studies of core formation inside molecular clouds, turbulence is the dominant ingredient. The usual strategy is the simulation of a periodic box of driven turbulence, optionally with or without self-gravity and magnetic fields (e.g., Padoan et al., 2007; Schmidt et al., 2010). Smith et al. (2009) derive the CMF from an non-driven, collapsing cloud. Although there are some subtleties in defining a core in such a simulation, the mass functions that they find are generally in good agreement with observations and theory.

The second part of the problem is arguably the more problematic one. The formation of individual stars inside cores has therefore been simulated in different contexts, from low- to high mass star formation and including various types of physical processes and feedback mechanisms, (e.g., Commerçon et al., 2011b; Commerçon et al., 2010; Hennebelle et al., 2011; Myers et al., 2013; Stamatellos et al., 2011; Walch et al., 2012) are just a few of the more recent studies. They generally find that fragmentation of cores is controlled

by a complex interplay of the physical processes involved, and that it is not possible to assign a dominant role to only one or two of those processes.

The partition of star formation into a "cloud to core" and a "core to star" problem is based on the assumption that stars accrete their mass predominately from the core in which they form. In situations where cores are not isolated but interacting with each other and possibly undergoing mergers, the above strategy looses its justification. Consequently, one must try to solve the problem in one piece by simulating star formation directly from turbulent molecular clouds. Up to date, all attempts to do so are based on limiting the dynamic range, either by decreasing the size of the setup or increasing the the size of the smallest resolution element. Studies that aim for sampling the IMF down into the brown dwarf regime usually choose an isolated $100\,\mathrm{M}_\odot - 1000\,\mathrm{M}_\odot$ molecular cloud as initial setup. The main picture that has emerged from such simulations (Bate, 2009, 2012; Girichidis et al., 2011; Krumholz et al., 2012; Myers et al., 2011) is that the number of brown dwarfs is overestimated when an isothermal or a barotropic equation of state is used. However, the aforementioned results obtained for individual cores suggest that the effects of radiation might be overestimated because of the absence of magnetic fields.

4.1. Isotropic infrared feedback

A mentioned in the introduction of this chapter, a consensus has emerged in recent years concerning the fact that some kind of radiative feedback from the protostars or the collapsing gas surrounding the protostar has to be included in simulations to obtain a realistic IMF. The assumption of isothermal gas is found to produce bottom-heavy IMFs (e.g. Girichidis et al., 2011). Calculations that solve the equations of radiation hydrodynamics (RHD) using the flux-limited diffusion approximation (Minerbo, 1978) have been shown to successfully reproduce the IMF over a large mass range (Bate, 2012; Krumholz et al., 2012). However, these calculations are computationally very expensive: the simulation reported by Bate (2012) for example consumed 6.4 million CPU hours during 34 months for the simulation of a $500\,\mathrm{M}_\odot$ molecular cloud! As impressive the results are, it is impossible to run large suites of these simulations to for example systematically test the influence of the physical environment on the IMF and thus address the question of its universality.

We aim for a computationally less involving way of incorporating radiative feedback from accreting protostars. Our method is based on the approximation described by Stamatellos et al. (2007). We assume the cloud to be optically thin, meaning that the mean free path of an infrared photon is much larger than the size of the cloud. Gas and dust are assumed to be perfectly coupled and in thermodynamical equilibrium with the radiation field. The sink particles act as the exclusive sources of radiation. Their luminosities

$$L_{\mathrm{sink}} = \epsilon_{\mathrm{IR}} \frac{G M_{\mathrm{sink}} \dot{M}_{\mathrm{sink}}}{R_{\mathrm{proto}}} \tag{4.1}$$

are determined by their mass M, accretion rate $\dot{M}$ and the assumed protostellar radius $R_{\mathrm{proto}} = 5\,R_\odot$.[1] The feedback efficiency ϵ_{IR} controls at which ratio the gravitational energy of the infalling gas is turned into infrared radiation. The sink particles emit radiation isotropically. In the optically thin limit, photons emitted by a isotropic source simply

[1] 1D RHD calculations which resolve the second collapse find values in this range for the radius of a protostar (Masunaga & Inutsuka, 2000).

stream to all directions at the speed of light. The radiation energy density E at distance r from the source is thus

$$E = \frac{L_{\rm sink}}{4\pi r^2 c}. \tag{4.2}$$

For the temperature of dust grain in thermodynamical equilibrium with the radiation field we have $aT^4 = E$, where a is the radiation constant. Adding the radiation energy densities from multiple sinks and an energy density of a background radiation field corresponding to a temperature $T_{\rm min}$ yields the total energy density in the radiation field. We solve for the temperature and obtain the temperature of the gas/dust fluid at location $\boldsymbol{r}$

$$T(\boldsymbol{r}) = \left(T_{\rm min}^4 + \sum_{i \in \text{sinks}} \frac{L_i}{16\pi\sigma_{\rm SB}|\boldsymbol{r} - \boldsymbol{r}_i|^2} \right)^{1/4}, \tag{4.3}$$

where $\sigma_{\rm SB}$ is the Stephan-Boltzmann constant. We usually assume the equilibrium temperature to be at 10 K. We denote this as the isotropic infrared (IIR) feedback model. It is obviously inaccurate when the optical depth of the gas surrounding a source approaches unity. In this case, self-shielding effects become important. The optically thick regions thus have to be accounted for with a different method. In Section 1.3.2 we have seen that for the collapse of a representative prestellar core, the temperature starts to raise quickly once a central density of $\sim 10^{-13}\,\mathrm{g\,cm^{-1}}$ is reached. This transition does naturally introduce a fragmentation scale. By adjusting the sink particle threshold density to $10^{-13}\,\mathrm{g\,cm^{-1}}$ and the sink accretion radius to the Jeans length and the density at this temperature, we introduce this fragmentation scale in our simulation.

The main approximations made in the IIR model are quite radical: no radiation anisotropy, no self-shielding, no compressional heating of the gas resulting in emission of radiation from extended sources. Rather than discussing possible implications of these approximations at length, we compare the IIR model to a RHD calculation which does not rely on any of them.

4.2. Comparing isotropic feedback to RAMSES-RT

In this section we compare our IIR model to a full RHD calculation. As reference scheme, we use the RAMSES-RT code. RAMSES-RT closes the equations of radiative transfer using the M1-approximation of the Eddington tensor (Levermore, 1984) and solves the resulting differential equations using a first-order Godunov scheme (Rosdahl et al., 2013). RAMSES-RT has recently been developed further to incorporate the treatment of optically thick regions (Rosdahl et al., in prep.) using the so-called "Isotropic Diffusion Source Approximation" introduced by Liebendörfer et al. (2009). In this method, the total radiation energy density is divided into two species of photons, the free-streaming and the trapped photons. The partition is based on the optical depth of an AMR cell which allows photons to change from one group into another when they leave or enter an optically thick region. In the RAMSES-RT runs, infrared emission from sinks as well as dust emission from hot gas, dust absorption and scattering is taken into account. The emission/absorption and scattering cross-sections are eye-fitted to the Planck/Rosseland mean opacities in the

Table 4.1: Simulation parameters for the disk accretion tests.

Sphere radius	R	$= 2000\,\mathrm{AU}$
Total gas mass	M	$= 4\,\mathrm{M}_\odot$
Density profile	$\rho(r)$	$= \frac{\rho_0}{(r/r_0)^2+1}$
	ρ_0	$= 4.7\times10^{-13}\,\mathrm{g/cm^3}$
	r_0	$= 10\,\mathrm{AU}$
Isothermal sound speed	c_s	$= 1.88\times10^{4}\,\mathrm{cm/s}$
$E_\mathrm{therm}/E_\mathrm{grav}$	α	$= 0.06$
Angular velocity	ω	$= 7.04\times10^{-12}\,\mathrm{s^{-1}}$
$E_\mathrm{rot}/E_\mathrm{grav}$	β	$= 0.5$
Box size	L_box	$= 32\,000\,\mathrm{AU}$
Cell size at levelmax	Δx_min	$= 7.8\,\mathrm{AU}$
Sink accretion radius	R_acc	$= 4\Delta x_\mathrm{min}$
Sink threshold	ρ_sink	$= 1.0\times10^{-13}\,\mathrm{g/cm^3}$

relevant temperature range (10 K, 1000 K) reported by Semenov et al. (2003),

$$\kappa_\mathrm{Planck} = \begin{cases} 0.1\,\frac{\mathrm{cm^2}}{\mathrm{g}} \times \left(\frac{T}{10\,\mathrm{K}}\right)^{1.7} & \text{if} \quad T \le 100\,\mathrm{K} \\ 5.0\,\frac{\mathrm{cm^2}}{\mathrm{g}} \times \left(\frac{T}{100\,\mathrm{K}}\right)^{-0.33} & \text{if} \quad T > 100\,\mathrm{K} \end{cases} \tag{4.4}$$

$$\kappa_\mathrm{Rosseland} = \begin{cases} 0.035\,\frac{\mathrm{cm^2}}{\mathrm{g}} \times \left(\frac{T}{10\,\mathrm{K}}\right)^{1.9} & \text{if} \quad T \le 100\,\mathrm{K} \\ 2.8\,\frac{\mathrm{cm^2}}{\mathrm{g}} & \text{if} \quad T > 100\,\mathrm{K}. \end{cases} \tag{4.5}$$

Due to angular momentum conservation of the infalling gas, accreting protostars are usually surrounded by accretion disks. Radiative feedback is believed to reduce or completely inhibit fragmentation of gas in the accretion disk. We thus compare the stabilizing effect of radiation on an accretion disk that forms during the collapse of a dense, rotating 4 $\mathrm{M}_\odot$ core. The parameters for this setup are listed in Table 4.1. We perform the comparison of ramses-rt (acronym RRT) and isotropic feedback (IIR) at two feedback efficiencies, $\epsilon_\mathrm{IR} = 1.0$ and $\epsilon_\mathrm{IR} = 0.1$, totaling to four runs which are labeled accordingly.

4.2.1. Results

In all runs a first sink forms at the center of the sphere right after the beginning of the simulation. We follow the evolution for 50 kyr which is approximately the duration of two rotations at the initial angular velocity. During this time $\approx 33\%$ of the total gas is transformed into sink particles. The final snapshots are shown in Figure 4.1. The most remarkable fact seen in this plot is the dependency of the number of sinks formed on the feedback efficiency ϵ_IR when we use the IIR model. At 100% efficiency, the feedback entirely prevents fragmentation of the disk, while at 10% efficiency, 7 sinks form in the disk. The number of fragments is less dependent on ϵ_IR in the RRT cases, in fact 9 sinks form inside the disk for both runs, which is not too different from the number of fragments

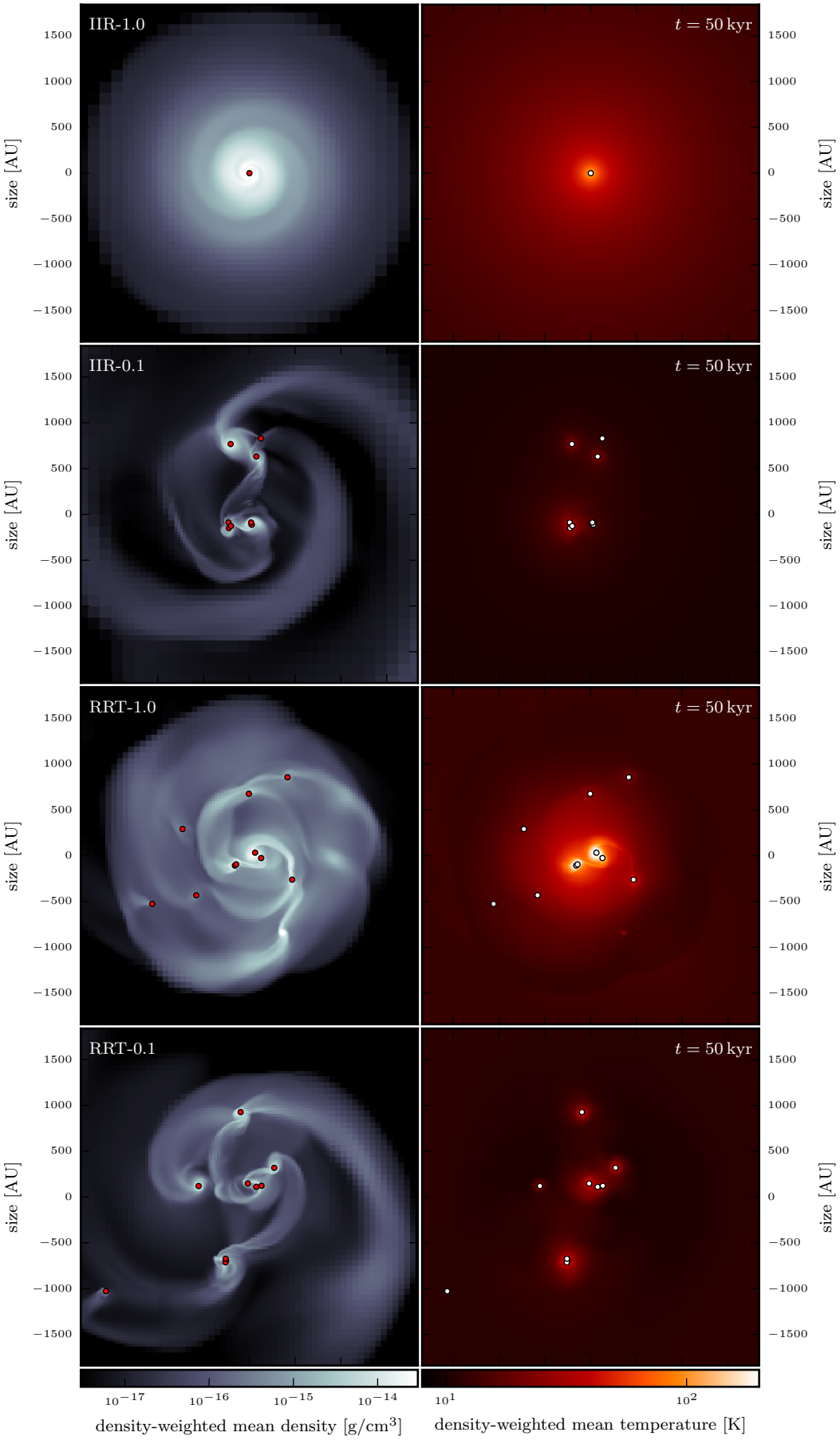

IIR-1.0
IIR-0.1
RRT-1.0
RRT-0.1
$t = 50$ kyr
size [AU]
1500
1000
500
0
−500
−1000
−1500
10^{-17}
10^{-16}
10^{-15}
10^{-14}
density-weighted mean density [g/cm^3]
10^{1}
10^{2}
density-weighted mean temperature [K]

Figure 4.1

Figure 4.1: Face-on view of the disk in our radiative transfer comparison case, taken after 50 kyr. Sink particles are marked with red and white dots in the left and right column respectively. The size of the dots indicates the sink accretion radius. The fragmentation behavior of the disk is less sensitive to the efficiency of the infrared feedback in the run performed with ramses-rt.

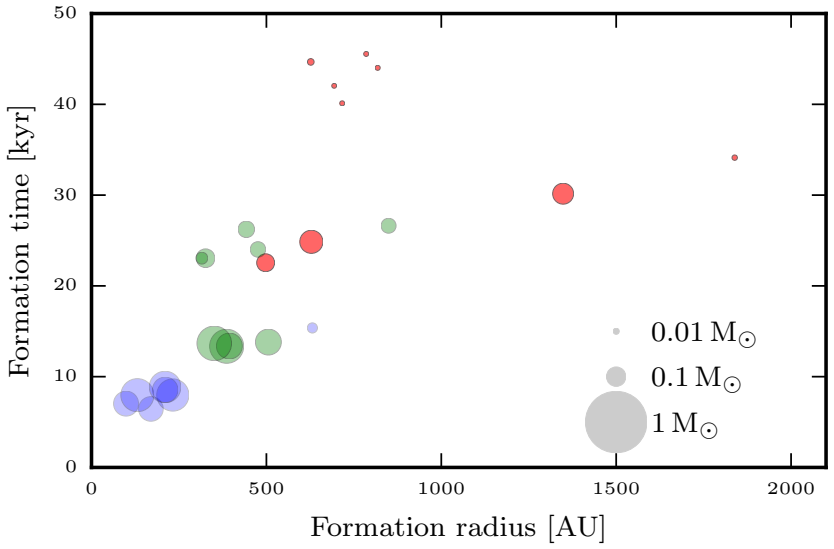


Figure 4.2: Formation time and radius of the sinks formed through fragmentation of the accretion disk. The size of the circles indicates the mass after 50 kyr. *Blue markers:* IIR-0.1 *Green markers:* RRT-0.1 *Red markers:* RRT-1.0

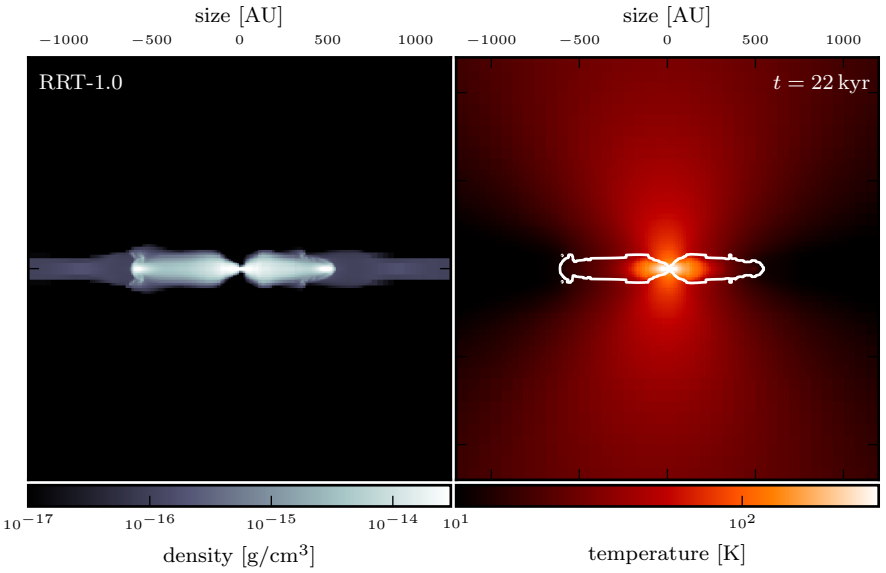


Figure 4.3: Slice through the disk taken immediately before the formation of the second sink (the first one from the fragmenting disk) for the RAMSES-RT run with 100 % feedback efficiency. The left panel shows a density slice and the right panel depicts a temperature slice with the white line indicating the $\rho = 1.0 \times 10^{-15}\,\mathrm{g/cm^3}$ contour from the density slice. The outer edge of the disk is shielded by its inner regions. The sink is going to form in the dense gas blob on the right hand side of the disk slice in the left panel at $\approx$ 500 AU distance from the center of the disk.

obtained in the IIR-0.1 case. Looking at the positions in the disk and the times at which the sinks formed, we however see a considerable difference. This is shown in Figure 4.2, which also indicates the mass of the sinks at $t = 50$ kyr, when we stop the simulations. The sinks in the IIR-0.1 run form early and close to the center of the disk. Hence, they can grow to a higher final mass. Consequently, they end up in small groups of sink. Note that for this test we did not allow sinks to merge.

The other end of the spectrum is marked by the sinks formed in the RRT-1.0 run, where sinks form later and further out in the disk. It is only at radii > 500 AU that the self-shielding of the disk becomes effective enough to allow fragmentation. The self-shielding and the resulting *torchlight* are shown in Figure 4.3, which shows an edge-on slice through the center of the disk and the gas blob which is about to trigger the formation of a first sink from the disk in the RRT-1.0 run. The effect of self-shielding can also be observed in the temperature-density phase diagrams of the gas (see Figure 4.4). The plot for the RRT-1.0 run at 20 kyr (lower-left panel) yields the presence of cold, dense gas which is inexistent in the corresponding IIR-1.0 run. Comparing the two right-hand panels which compare the cases with a reduced feedback efficiency at an earlier stage shows another effect: here it is the lack of compressional heating and the inability to model an optically thick region that allows the disk to stay relatively cool in the IIR case. This picture emerges even clearer from the temperature profiles at $t = 6$ kyr (Figure 4.5). The disk is coolest in the IIR-0.1 case which leads to fragmentation shortly after at $\approx$ 200 AU. The difference in temperature observed outside the disk between the IIR cases and the corresponding RRT cases are caused by the additional infrared emission from the compressionally heated gas.

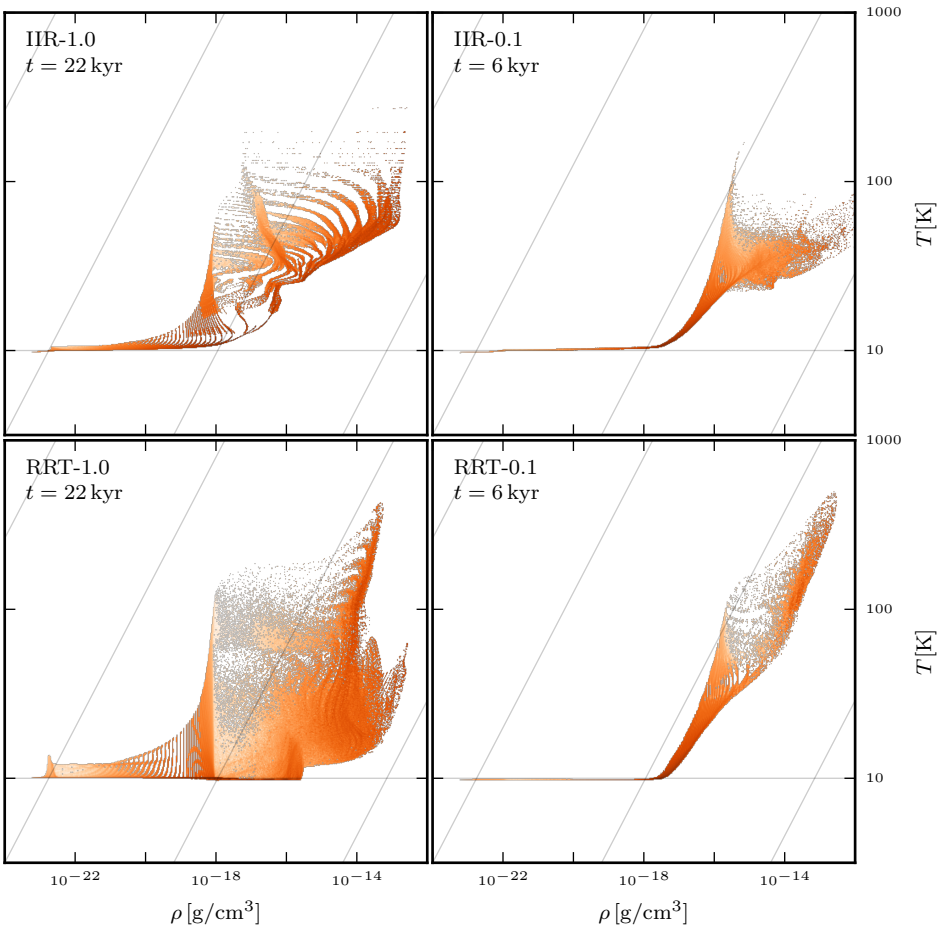


Figure 4.4: Phase diagrams for the accretion disk simulations. The colormap indicates the logarithmic mass per bin in arbitrary units. The horizontal gray line indicates the temperature floor of 10 K and the slope of the ascending lines correspond to a power-law index of 0.4. Note that all histograms were computed for snapshots where there was only one sink present. The two left hand panels show the status shortly before the formation of the first sink in the RRT-1.0 run. The sink will form in the cold high-density gas which is not present in the IIR-1.0 run. The right-hand panels depict the situation just before fragmentation of the disk in the IIR-0.1 run.

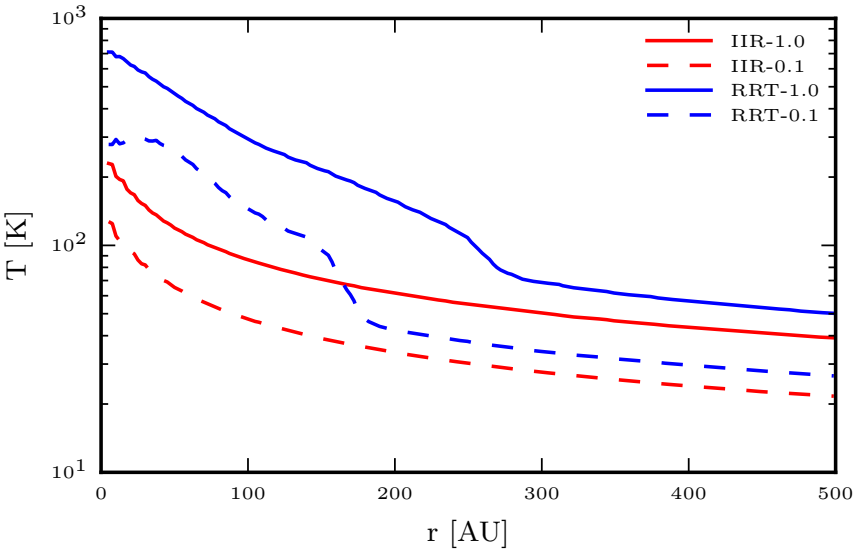


Figure 4.5: Temperature profiles of the disks at $t = 6\,\mathrm{kyr}$, briefly before the first disk fragment forms in the IIR-0.1 case at $r \approx 200\,\mathrm{AU}$. In the RRT cases, the drop seen in the temperature profile coincides with the edge of the disk.

The accretion luminosities of the sinks are similar in the corresponding IIR and RRT cases and cannot explain the discrepancy.

4.2.2. Conclusions

The most obvious conclusion from this experiment is that self-shielding effects are very important in calculations that involve possibly fragmenting accretion disks. This allows fragmentation to occur relatively close to luminous protostars; this is not observed when the feedback energy carried by infrared photons is streaming freely in all directions. This effect is possibly very important for simulations trying to reproduce the observed IMF. Many simulations of isolated cores have demonstrated this effect before (e.g., Krumholz et al., 2007). To what extent the details of this effect are determined by the chosen radiative transfer scheme (flux limited diffusion or M1) is a question that is worth considering in the future.

For our isotropic feedback model, the comparison has the following implication: using it at full feedback efficiency is very likely to overestimate the effect of radiative feedback. Reducing the feedback efficiency can result in a similar number of fragments but is unlikely to recover all observed properties of the sinks formed in a full RHD calculation. We see two effects with opposing impacts on the final number of fragments formed in the disk. *i)* At early time and at short distance, full RHD prevents fragmentation in the compressionally heated, optically thick disk (Figure 4.5). *ii)* At late time and at large distance, full RHD allows more fragmentation in the self-shielded region of the disk (Figures 4.1, 4.3).

Stamatellos et al. (2011) reported on a similar study where an isotropic feedback model completely prevented fragmentation in an accretion disk. Instead of uniformly reducing the feedback efficiency, they modeled episodic accretion where sharp spikes in the accretion rate are followed by phases of only little accretion. The interim phases of low accretion rates and thus low accretion luminosities allowed the disk to fragment. This might be

another technique worth exploring in the future. It should also be mentioned that the picture that emerges from our test possibly changes when a magnetized core is considered instead. Magnetic fields might prevent the formation of a thin, massive disk in the first place and thus reduce the anisotropy in the radiation field around accreting protostars. It is thus presumably the interplay of radiation and magnetic fields that controls fragmentation (Commerçon et al., 2011b).

Another conclusion that we derive from this experiment is the feasibility of star cluster formation simulations using RAMSES-RT. Directly comparing the computational expense of the different runs is difficult as they were performed on different systems using a different number of CPUs. However, the time step in the RAMSES-RT runs is typically two orders of magnitude smaller than the time step in the IIR runs. The small time step seen in the RAMSES-RT runs is caused by the relatively high *reduced light speed* that has to be used when regions of high optical depth are present. We can thus estimate that RAMSES-RT will slow down a calculation by a factor of 100 as soon as regions of $\sim 10^{-13}\mathrm{g\,cm^{-3}}$ and $\sim 100\,\mathrm{K}$ are resolved. This is a strong incentive to use the IIR model once we want to go to larger setups but keep the resolution at AU scale.

4.3. A Lagrangian refinement strategy

When studying resolution effects on AMR simulations, it makes sense to distinguish two different questions. First, one can ask how the choice of the smallest grid spacing allowed in the simulation affects the results. Answers to that question are ideally independent of the AMR technique and thus valid also for simulations using a fixed cartesian mesh. The other question is the one we want to address in this section: how does de-refining parts of the computational volume and updating them using a coarser spatial resolution influence the outcome of the simulation? For relatively small setups one can follow the strategy that we have already employed in Section 2.4 and simply compare to a fully refined reference case. This is not possible for most scientifically relevant AMR simulations.

For the case of gravitational collapse in self-gravitating gas there is one well accepted refinement criterion that is routinely used. It was established by Truelove et al. (1997), who found that the local Jeans length needs to be resolved by at least four grid cells to avoid artificial fragmentation. Recalling the role of the Jeans length as the largest scale that is supported against collapse by thermal pressure only (see Section 1.2.2), resolving the local Jeans length does not only avoid artificial fragmentation, it should also make sure that no fragmentation at unresolved scales is missed. The Truelove criterion furthermore guarantees that the gas mass inside one cell does not exceed $\approx 1/32$ of the local Jeans mass which is thus resolved too.

A condition similar to the Truelove criterion has been introduced as a resolution requirement for SPH simulations by Bate & Burkert (1997) who found that the local Jeans mass should be resolved by $2N_{\mathrm{neigh}}$ SPH particles, where N_{neigh} denotes the number of neighbors considered in the SPH smoothing kernel. Unless particle splitting (Kitsionas & Whitworth, 2002) is employed, the mass resolution in an SPH simulation is fixed for the entire run. It is thus the *smallest* Jeans mass[2] that is resolved throughout the entire calculation. This condition can be turned into a Lagrangian refinement strategy for AMR simulations. This strategy consist in refining a cell as soon as the gas mass inside the cell

[2]By this term we mean the smallest Jeans mass that will show up at any point during the calculation. For an isothermal run employing sink particles, this is simply the Jeans mass at the sink threshold density.

surpasses a fraction of a fixed mass which is chosen to be the smallest Jeans mass showing up in the simulation. The difference between the Truelove criterion and the Lagrangian strategy thus boils down to the question whether the *local* or the *smallest* Jeans mass is resolved during a calculation. This distinction has a big effect on the computational costs of the simulation. For example, an isothermal collapse calculation covering six orders of magnitude in density translates into three orders of magnitude between the initial and the smallest Jeans mass. Employing the Lagrangian strategy therefore increases the initial number of cells in the calculation by a factor of 1000. Even at later times, only a small fraction of the gas is effectively undergoing collapse at the same moment and the number of leaf cells in the entire calculation can thus be decreased greatly by using the original Truelove criterion.

In the above discussion, effects of the refinement strategy on the properties of the velocity field have been completely neglected. In Section 2.5 we have seen that the details of the refinement criterion can influence the turbulent cascade from large to small scales, even if a fixed number of computational cells is employed. We can thus expect that increasing the number of leaf cells by orders of magnitudes has a huge impact on velocity fluctuations and velocity induced density fluctuations at small scales. In this section, we study whether these additional small scale fluctuations can trigger the formation of additional sink particles or not.

Another possibly important role in the above considerations is played by the sound scale. At scales larger than the sound scale, velocity fluctuations are supersonic and provide more support against gravity than thermal pressure does (see Section 1.2.3.1). In the particular setup that we have used for this comparison, the Lagrangian criterion is more strict and hence ensures that the sound scale is properly resolved. Since the initial Jeans length is slightly above the sound scale, this is not the case when we use the Truelove criterion. However, for the initial Jeans length to be significantly larger than the sound scale, a bigger and hence less dense cloud would be necessary. We can therefore not use our test setup to examine how the question of whether the sound scale is resolved or not, will affect the results. Consequently, we remove this effect completely from the analysis by choosing the base grid such that the sound scale is always resolved, no matter if we are using the Truelove or the Lagrangian criterion.

4.3.1. Setup

We test the role of the refinement criteria simulating the collapse of a $100\,\mathrm{M}_\odot$ molecular cloud. The numerical parameters of the setup can be found in Table 4.2. The cloud is initially spheroidal in shape with axis ratios of 2:1:1 and the density distribution is homogenous. The velocity field is initialized with a velocity power spectrum $P_v(k) \propto k^{-4}$ and contains both, solenoidal and compressive modes. The velocity field is normalized such that the kinetic energy matches half the gravitational energy. This results in a sound scale of $\lambda_{\mathrm{sound}} = 0.028\,\mathrm{pc}$ which is considerably smaller than the usual 0.1 pc found in Section 1.3.1. This is because of the cloud size which is at the lower end of the observed values for the given mass. A sphere big enough to contain supersonic internal velocity does therefore have only a little more than four cells in diameter initially. However, as the turbulence level decays during the calculation, we conclude that the sound scale is sufficiently resolved in the run where we use the Truelove refinement criterion. This claim is strengthened by the fact that in both runs, groups of sinks form at very similar locations. Sinks are used as

Table 4.2: Physical and numerical parameters of the refinement criterion comparison.

Radii	$R_1 : R_2 : R_3 = 0.70 : 0.35 : 0.35\,\mathrm{pc}$
Mass	$M = 100\,\mathrm{M}_\odot$
Density	$\rho = 1.9 \times 10^{-20}\,\mathrm{g/cm^3}$
Free-fall time	$t_\mathrm{ff} = 4.83 \times 10^5\,\mathrm{yr}$
Mean molecular weight	$\mu = 2.33$
Temperature	$T = 10\,\mathrm{K}$
Isothermal sound speed	$c_s = 1.88 \times 10^4\,\mathrm{cm/s}$
$E_\mathrm{therm}/E_\mathrm{grav}$	$\alpha = 0.07$
$E_\mathrm{kin}/E_\mathrm{grav}$	$\beta = 0.5$
Turbulent mach number	$\mathcal{M}_\mathrm{rms} = 4.0$
Sound scale	$\lambda_\mathrm{sound} = 0.028\,\mathrm{pc}$
Box size	$L_\mathrm{box} = 3.14\,\mathrm{pc}$
Maximum level of refinement	$l_\mathrm{max} = 18$
Cell size at levelmax	$\Delta x_\mathrm{min} = 2.47\,\mathrm{AU}$
Minimum level of refinement	$l_\mathrm{min} = 8$
Cell size at levelmin	$\Delta x_\mathrm{max} = 0.012\,\mathrm{pc}$
Sink accretion radius	$R_\mathrm{acc} = 4\Delta x_\mathrm{min}$
Sink threshold	$\rho_\mathrm{sink} = 1.0 \times 10^{-13}\,\mathrm{g/cm^3}$
Mass resolution element	$m_\mathrm{Lag} = 1.35 \times 10^{-5}\,\mathrm{M}_\odot$

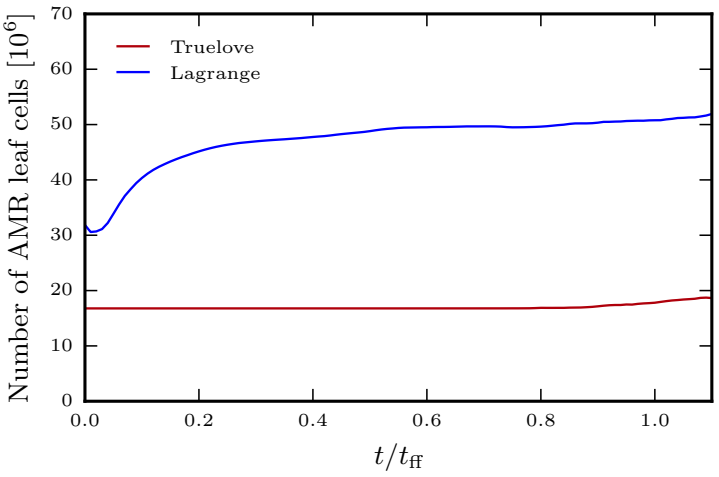


Figure 4.6: Number of AMR leaf cells during the simulation for the tested refinement criteria. The number of leaf cells for in the Truelove case is mainly set by the $256^3 \approx 16{\times}10^6$ base grid.

described in Chapter 3. Note that they are allowed to merge during their first core lifetime which is assumed to last for 5 kyr.

We expect the additional small scale fluctuations to influence the fragmentation of cores into individual protostars. This fragmentation should also be affected by radiative feedback from the first collapsing fragment. We therefore apply both refinement criteria to an isothermal case and a case with isotropic infrared feedback from the sink particle with an efficiency of $\epsilon_{\rm IR} = 0.1$ (see Section 4.1). The combination of original Truelove and Lagrangian refinement criterion with isothermal gas (acronym ISO) and infrared (acronym IR) feedback leads to a total of four runs in the comparison.

4.3.2. Results

We now turn to the results of this experiment. For plots and figures which are indistinguishable for the respective ISO and IR runs, we only show one version (i.e. the IR one) per refinement strategy. In Figure 4.6 we plot the number of AMR leaf cells during the simulation for the Truelove and the Lagrange case. The Lagrangian criterion increases the number of leaf cells by about a factor of three. For our particular setup, the difference in the computational costs is much larger, because in the Truelove case a higher fraction of cells occupy lower AMR levels which are sub-cycled since we use adaptive time stepping. The number of leaf cells in the Truelove case would be even smaller if the code was enforced to keep the minimum level of refinement at Level 8. The big difference in terms of refinements is also shown in Figure 4.7, where we show the maximum level of refinement along the line-of-sight for a view on the main filament forming in the simulation. For most of the uncollapsed cloud regions, there are approximately three levels of difference in refinement. Because of the fixed required mass resolution in the Lagrangian case, the refinement levels effectively mark density contours of the gas. If the Truelove criterion is applied, refinements are very localized around collapsed region with multiple refinement boundaries within a short distance. The discrepancy in resolution leads to a very different density structure inside the cloud on small and intermediate scales. This is shown in Fig-

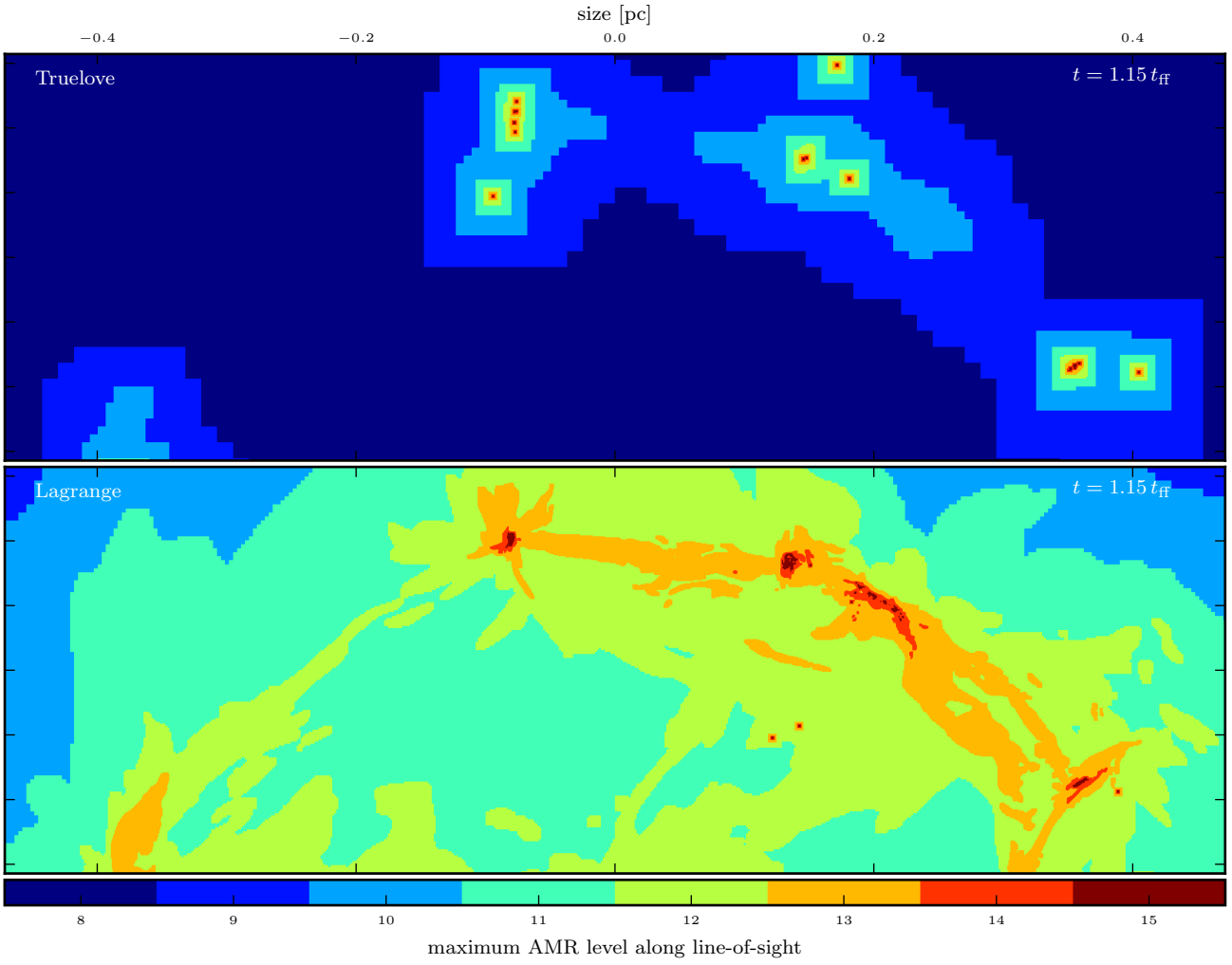


Figure 4.7: Refinement maps corresponding to the density snapshots shown in Figure 4.8. Note that the maximum level of refinement in this run is 18. Those regions would be barely visible in this plot and are thus just considered being at level 15. Sinks automatically trigger refinement of their host cells to the highest level.

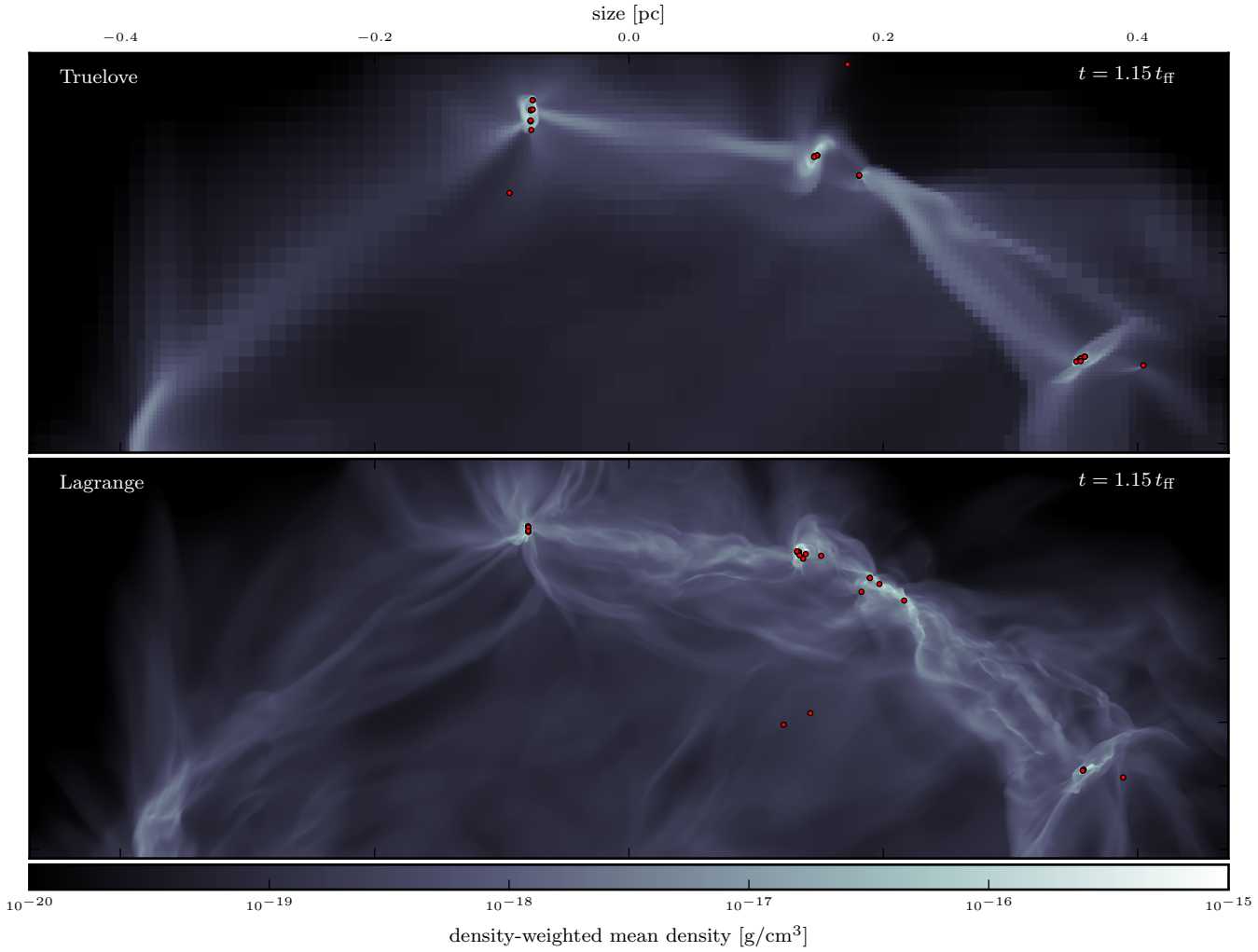


Figure 4.8: Gas density structure of the same region that is shown in Figure 4.7. Positions of sink particles are over plotted as red dots. In order to be visible, the size of the dots does not represent the true size of the sink accretions zone.

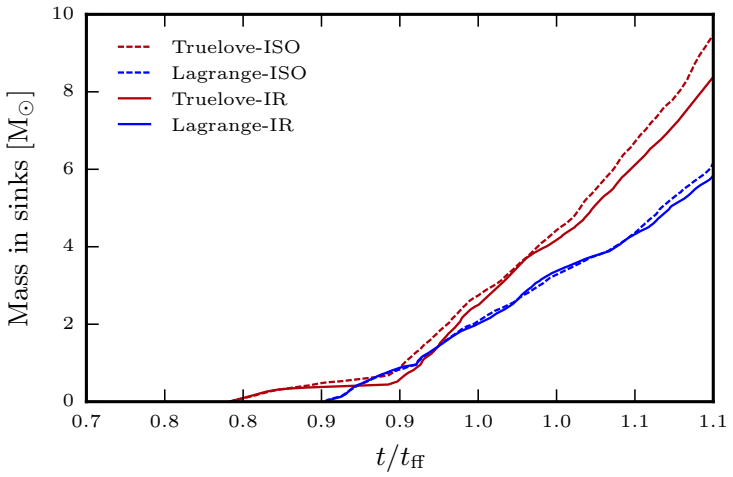


Figure 4.9: Total mass in sinks as a function of time. The additional small scale fluctuations in the "Lagrange" runs slow sink formation and accretion on sinks.

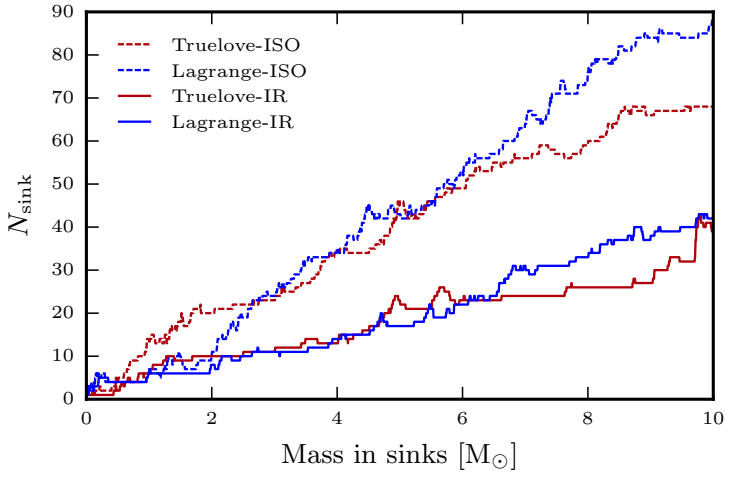


Figure 4.10: Number of sinks in each run for a total mass in sinks. While the effect of infrared feedback on the number of sinks and thus the mean sink mass is obvious, the two refinement criteria lead to very similar results.

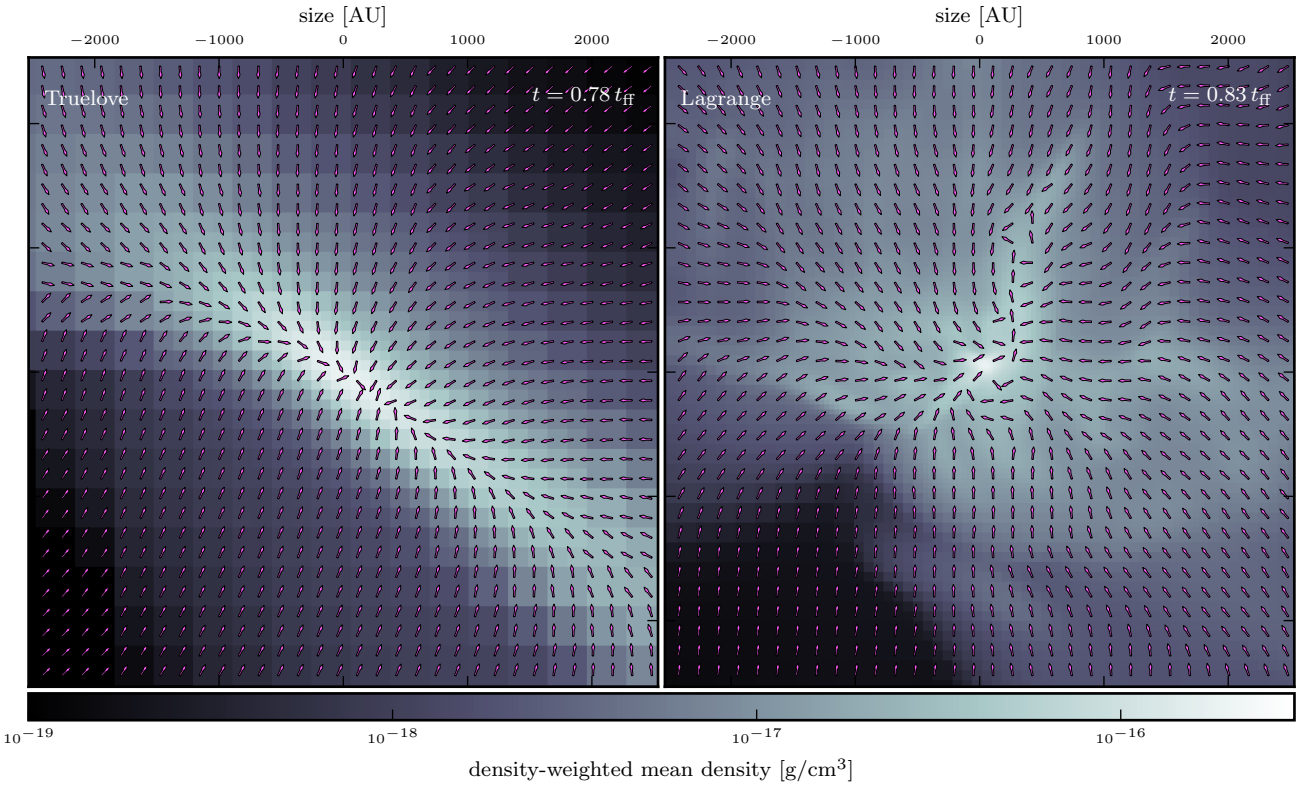


Figure 4.11: Snapshots of the first collapsing region for both refinement criteria, taken shortly before the first sink forms. The additional refinements caused by the Lagrangian criterion lead to the presence of small scale features which are inexistent in the Truelove run. The pink arrows indicate the direction of the density-weighted mean velocity projected onto the plane perpendicular to the line of sight.

ure 4.8. However, as it can be seen in the same plot, both setups trigger collapse and sink formation in the same locations. We assign this to turbulence producing the same cores as soon as the sound scale is resolved.

What is more surprising is the fact that the subsonic motions and the resulting density fluctuations seen in the image do not significantly change the total number of fragments formed. This is shown in Figure 4.10 where we plot the number of sinks for the different runs as a function of the total mass in sinks. In the isothermal case, the small scale density and velocity fluctuations do only mildly increase the number of formed sinks and thus decrease the average sink mass. This further diminishes when we use infrared feedback. The increased temperature adds thermal support to the observed density fluctuations which does not allow them to form additional sinks. Our merging strategy to merge young sink particles is likely to contribute to this results, as in the Lagrange run, the ratio of totally formed sinks to surviving sinks is slightly higher than in the Truelove run. This means that the small amount of additional sink formation in the Lagrange runs is generally happening close to pre-existing sinks. A larger difference between the two refinement strategies is seen in the total mass in sinks as a function of time, plotted in Figure 4.9. This can be understood by taking a closer look at the first collapsing regions in both cases (see Figure 4.11). When only the Truelove criterion is used, the collapsing regions show no substructure once they have reached densities around $10^{-16}\,\mathrm{g\,cm^{-3}}$ and the gas is moving towards the center of the collapse homogeneously. In the case of the Lagrangian criterion, the gas still shows signs of subsonic turbulence with non-zero angular momentum which needs to dissipate to allow further collapse.

4.3.3. Conclusions

In this section we have performed an experiment to test effects of the chosen refinement criterion in molecular cloud simulations. We resolved the initial sound scale and tested how additional refinements triggered by a Lagrangian criterion or the original Truelove criterion affect the number and properties of the sink particles formed. We have found that the results are surprisingly robust and influenced only mildly by additionally resolved small-scale velocity and density fluctuations. Infrared feedback from accreting sinks locally increases internal support of over-dense regions due to thermal pressure. It thus further decreases the small difference in the number of fragments found for the isothermal runs. We therefore conclude that resolving the sound scale and the local Jeans length is sufficient for this kind of simulations. This allows a big speed gain. In our test, the run using the Lagrangian criterion was more than an order of magnitude slower than the Truelove run. To our knowledge, no such comparison has been performed so far. Our results are reassuring since the Truelove criterion has been the standard refinement rule for AMR simulations of self-gravitating gas. Furthermore, our Lagrangian criterion adapts the spatial resolution to the flow in the same way as SPH codes do. Our results therefore suggest that the different ways how SPH and AMR codes adapt the spatial resolution should not lead to a large discrepancy between the results obtained with the two methods.

The overall situation might be different for the magneto-hydrodynamics case. Federrath et al. (2011) found that 32 cells per Jeans length are necessary to properly resolve the amplification of magnetic fields inside a collapsing magnetized core. They claim that otherwise, turbulent energy, turbulent pressure and especially vortices are not sufficiently resolved. This corresponds well to our finding that the collapse of core-like structures is happening faster in the Truelove runs than in the corresponding Lagrange simulations.

Table 4.3: Physical and numerical parameters of the fragmenting 1000 $M_\odot$ molecular cloud runs.

Radii	$R_1 : R_2 : R_3 = 3.0 : 1.5 : 1.5\,\mathrm{pc}$
Mass	$M = 1000\,M_\odot$
Density	$\rho = 2.36 \times 10^{-21}\,\mathrm{g/cm^3}$
Free-fall time	$t_\mathrm{ff} = 1.37 \times 10^6\,\mathrm{yr}$
Mean molecular weight	$\mu = 2.33$
Temperature	$T = 10\,\mathrm{K}$
Isothermal sound speed	$c_s = 1.88 \times 10^4\,\mathrm{cm/s}$
$E_\mathrm{therm}/E_\mathrm{grav}$	$\alpha = 0.09$
$E_\mathrm{kin}/E_\mathrm{grav}$	$\beta = 0.5$
Turbulent mach number	$\mathcal{M}_\mathrm{rms} = 6.2$
Sound scale	$\lambda_\mathrm{sound} = 0.05\,\mathrm{pc}$
Box size	$L_\mathrm{box} = 8.62\,\mathrm{pc}$
Maximum level of refinement	$l_\mathrm{max} = 18$
Cell size at levelmax	$\Delta x_\mathrm{min} = 6.78\,\mathrm{AU}$
Minimum level of refinement	$l_\mathrm{min} = 8$ (9 on the volume of the ellipsoid)
Cell size at level 9	$\Delta x_\mathrm{max} = 0.017\,\mathrm{pc}$
Sink accretion radius	$R_\mathrm{acc} = 4\Delta x_\mathrm{min} \approx \lambda_\mathrm{Jeans}(\rho_\mathrm{sink}, c_s)$
Sink threshold	$\rho_\mathrm{sink} = 1.0 \times 10^{-13}\,\mathrm{g/cm^3}$

However, even when the local Jeans length is resolved by 32 or even more cells, the total number of resolution elements will increase as the gas is undergoing collapse. We thus suggest that for the sake of producing realistic small scale velocity fluctuations in collapsing regions, a Lagrangian refinement criterion (which corresponds to a higher number of cells per local Jeans length in low than in high density regions) might be a more promising strategy since it keeps the number of resolution elements roughly constant.

4.4. IMF from a fragmenting 1000 $M_\odot$ cloud

In this section we put the pieces together and simulate the formation of a star cluster in a $1000\,M_\odot$ molecular cloud. We use this to study the effect of the isotropic IR feedback on the resulting IMF. We therefore perform three simulations: an isothermal one where the gas temperature is kept fixed (acronym 10 K), one with isotropic IR feedback with an efficiency of 10% (acronym IR-0.1) and one with an efficiency of 100 % (acronym IR-1.0). The initial gas distribution is a homogeneous ellipsoid with aspect ratios of 2:1:1 and a long diameter of 6 pc. The initial turbulent velocity field is initialized with a velocity power spectrum $P(k) \propto k^{-4}$ and contains solenoidal and compressive modes. Further parameters describing the setup are found in Table 4.3. At this point we want to stress that our initial setup is still on the smaller side of the range found in observations (see Figure 1.2) but much more representative than the setups used for instance by Bate (2009, 2012) who start with a very high density cloud, presumably to reduce the necessary dynamic range

in the simulation. The choice of a prolate ellipsoid rather than a perfect sphere as a setup has to be seen in the same spirit: since we do not replenish dissipated turbulence during the simulation, the random motions will unavoidably dissipate and be replaced by a global contraction mode. An artificially symmetric setup might in this case lead to an extremely dense central region and thus extreme conditions for star formation. The sink formation threshold $\rho_{\rm sink} = 1.0 \times 10^{-13}\,{\rm g\ cm^{-3}}$ and accretion radius $R_{\rm acc} \approx 27\,{\rm AU}$ are setting the relevant scale for fragmentation. Our choice reflects the opacity limit and the corresponding Jeans length found in molecular gas at 10 K. Sink particles are assumed to represent first Larson cores undergoing Kelvin-Helmholtz contraction during the first 5 kyr after their formation. In this time, they are allowed to merge with other sinks (see Section 3.4.1). The sinks accrete their mass through "flux accretion" (Section 3.6.3) and the angular momentum of the accreted gas is assigned to the remaining gas in the accretion zone (Section 3.6.4). The accelerations of sink particles are computed using the PM method for sinks with a mass below $0.5\,M_\odot$ and by direct force summation for the massive sinks (Section 3.5).

4.4.1. Results

The presence of a compressive large-scale mode in the initial velocity field lets the cloud contract fastest along its long axis. This results in an overall compressed central region of $\approx 1\,{\rm pc}^3$ in the cloud rather than the formation of a filamentary structure that would be expected given the initial shape. However, our main goal of producing stars in a scattered rather than extremely centralized fashion is still obtained as can be seen by the snapshots shown in Figure 4.12. The first sinks form after $0.54 t_{\rm ff}$ in this region compressed by the initial turbulent motions. At this moment, there are approximately $150\,M_\odot$ of gas inside the central $1.0\,{\rm pc}^3$ which corresponds to roughly four times the initial density. Subsequently, most of the sinks form inside this central region which is depicted in Figure 4.12 at different times. The simulations are stopped when $\gtrsim 10\%$ of the initial mass has been turned into sinks.

The indistinguishable density structure of the gas between the different rows in the plot belonging to the different simulations shows that the large-scale evolution of the cloud is unaffected by the IR feedback which in turn controls fragmentation on a smaller scale. This results in a large discrepancy in the number of objects with a mass below $0.3\,M_\odot$ between the different runs. This is shown in Figure 4.13 which depicts the cumulative and differential mass sink mass functions at two different times in the simulation. The results of the IR-1.0 run confirm the picture that already emerged in Section 4.2: isotropic infrared feedback is too effective in preventing subsequent fragmentation. The intermediate scenario IR-0.1 is producing fragments in the right mass range. However, there is a deficit of objects with a mass $\sim 0.1\,M_\odot$. A possible explanation for this lies in the results of Section 4.2, especially Figure 4.2. In the simulations employing full RHD, fragments formed later and further out in the disk. Consequently, they will remain less massive and have a smaller probability to merge during their "childhood". This should overall lead to more low mass objects. Another possible explanation is of more dynamical nature. Sink-sink interactions are softened at the scale comparable to the grid spacing. This can artificially reduce the number of ejections and thus allow those objects to accrete longer than they should. The isothermal run produces too many brown dwarfs as expected. However, the excess is not as large as expected. As already shown Section in 3.7.2, this discrepancy to results found by other authors for isothermal runs (Girichidis et al., 2011; Myers et al., 2011) is caused

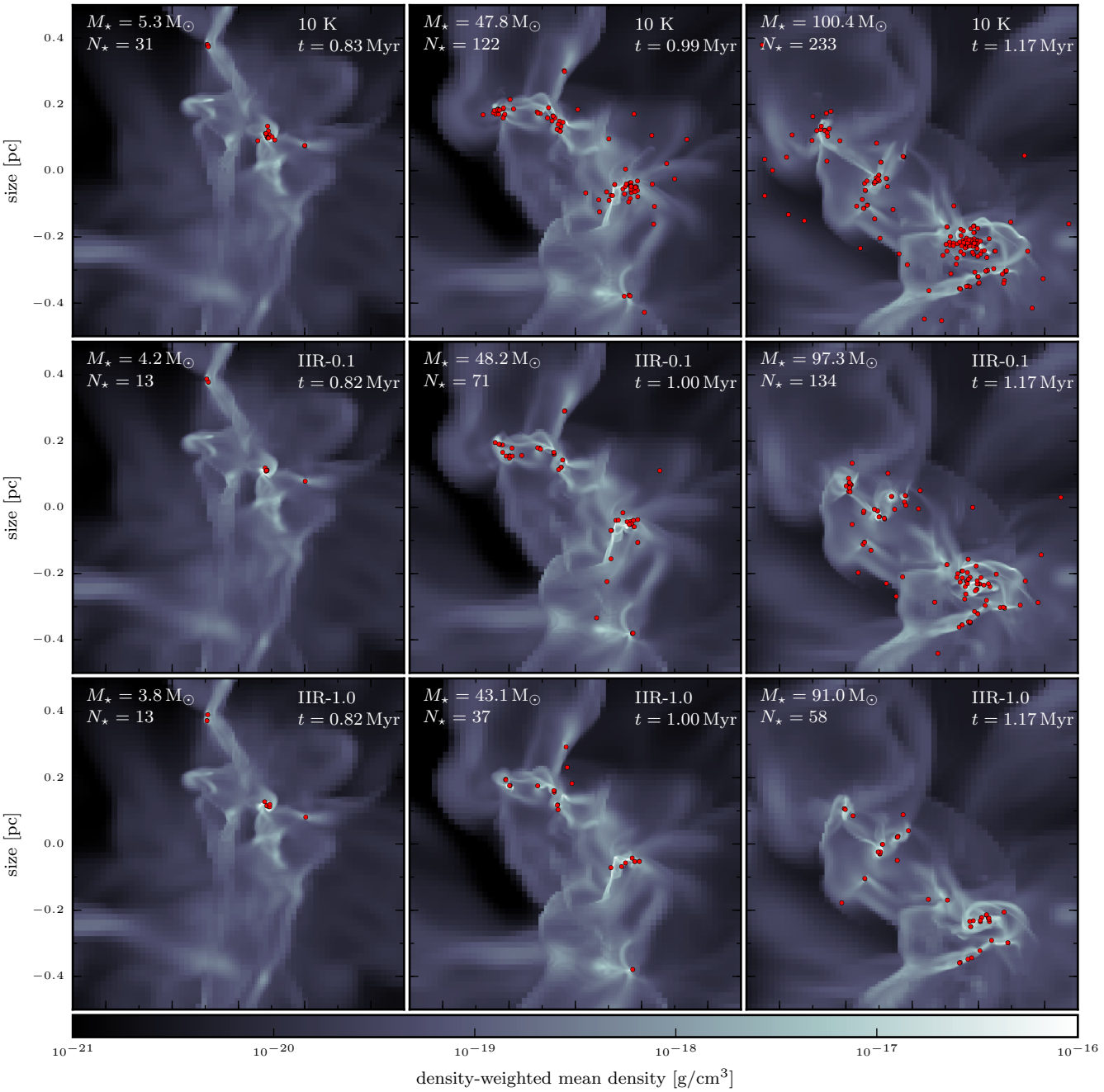


Figure 4.12: Temporal evolution of the main star forming region in the molecular cloud. Sink particles are indicated by red dots which are not to scale with the accretion radius. The number of sinks and the total mass in sinks that are contained *inside* the depicted volume are indicated in each panel. Note that many of the red dots do in fact represent multiple sink particles which are located too close to each other to be kept apart at the given scale.

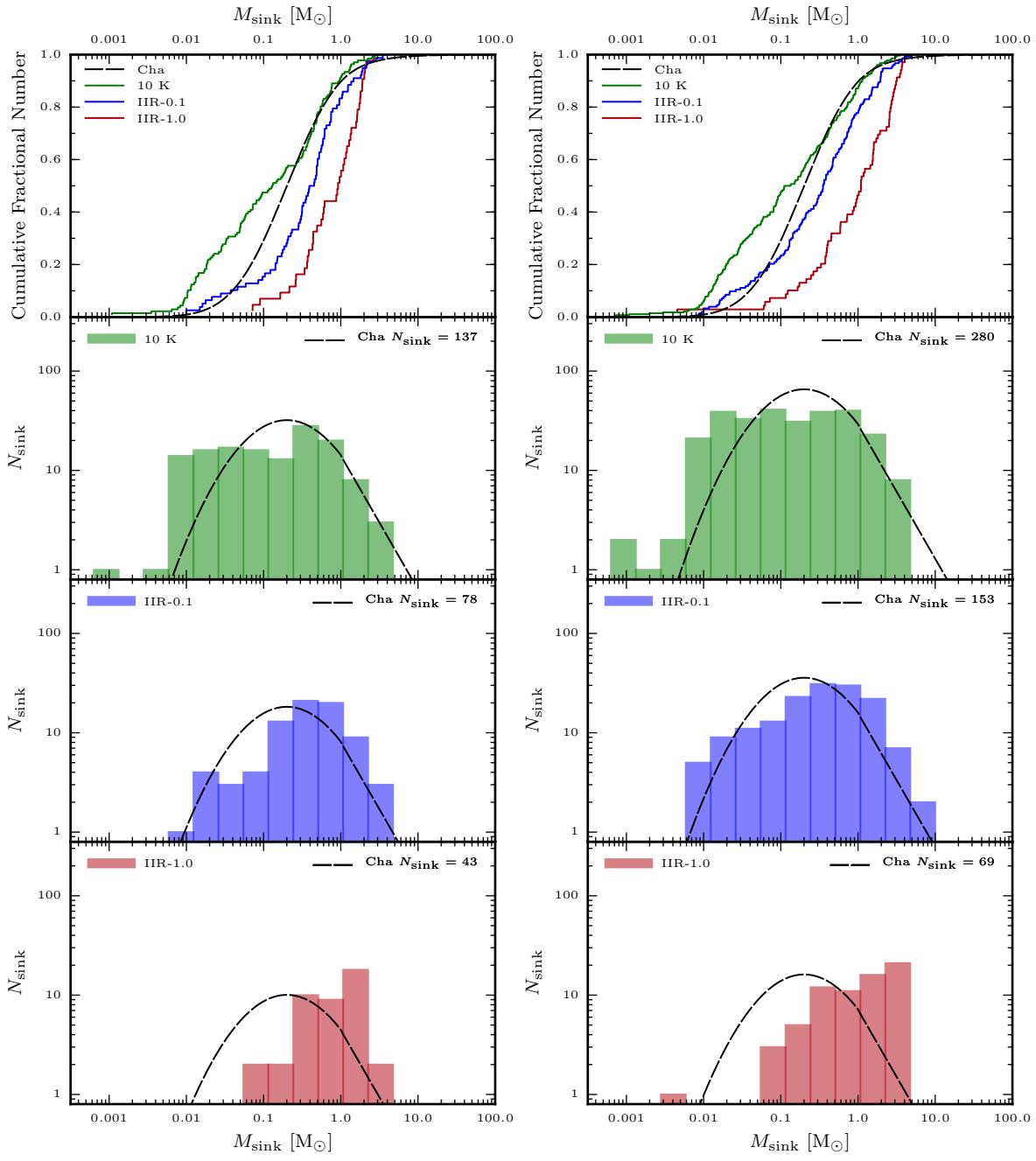


Figure 4.13: Massfunctions of the sink particles obtained with different IIR feedback efficiencies. The left column corresponds to the snapshots at $t = 1.0 \times 10^6$ yr in Figure 4.12. The right column represents the end of the simulations at $t = 1.17 \times 10^6$ yr. The dashed lines indicate the Chabrier (2005) mass function. In the histogram plots the mass function is normalized to match the total number of sink particles in the simulation.

by the details of the sink particle implementation. Compared to the experiment presented in that section, the temperature of the cloud described here is lower. On the other hand we have used sink merging which was turned off in the aforementioned case. This results in a overall very similar mass function of sink particles.

The high mass end of the sink mass function is similar to the observed IMF for the 10 K and the IIR-0.1 runs. It is remarkable that both mass functions obtain their characteristic slopes very early in the simulations (see left panel in Figure 4.13). This suggests that the masses in that range are imprinted already in the turbulence and not determined by sinks competing for the infalling gas. This interpretation of the results is somewhat conflicted by the sink masses seen in the IIR-1.0 run, especially at late times. If we assume that the over-efficient radiative feedback suppresses any secondary fragmentation of cores, then the sink mass function should be more similar to the observed clump mass function and at least produce a more realistic behavior at high masses.

4.4.2. Conclusions

Our IIR feedback model essentially contains two-parameters. The first parameter is the density at which sink particles are formed. We argue that this scale must be chosen according to the opacity limit in molecular clouds, but there remains some range in which this value can be varied. The second parameter is the discussed feedback efficiency. So far, we have only varied the latter of the two parameters, keeping the first one fixed at a value that we consider the most likely one. While a complete verification study of the model must clearly analyze effects of both parameters, the results obtained so far are quite promising. The main effect of radiation can be included in simulations simply by choosing the right scales for sink formation. Furthermore, our merging criterion reflects the slow contraction of gas in the optically thick regime. Consequently, we find the results of isothermal calculations to be less bottom-heavy than previous authors (Girichidis et al., 2011; Myers et al., 2011). Increasing the temperature in the vicinity of accreting sinks through radiative feedback has a dramatic effect on the number of observed sinks in the range $(0.01\mathrm{M}_\odot, 0.1\mathrm{M}_\odot)$. In fact, even our model with a reduced feedback efficiency seems to be too effective in reducing the number of low mass fragments. This agrees to our findings from the isolated disk experiment. Comparable calculations which perform the full RHD predict a more realistic IMF in that range (Bate, 2012; Krumholz et al., 2012).

Star forming cores are well scattered throughout the cloud. Nevertheless, the characteristic slope in the high mass tail of the IMF develops quickly for the two more realistic runs in our comparison. This points towards a turbulent origin of the IMF, where the mass of the star is mainly determined by the core where it is formed. This was very convincingly shown by Smith et al. (2009), who compared the mass of a sink directly to the mass of the core is has emerged from. Unless for our less realistic case with full feedback efficiency, the mean mass of a sink remains relatively constant and we do not see evidence for overheating as it was reported by Krumholz et al. (2011). We assign this to the comparably low density initial conditions that we have set up and the resulting star formation which is less concentrated than in the simulation of Krumholz et al. (2011).

One of the calculations described above can be performed in roughly one week on 128 CPUs. This enables us to perform many of those simulations for one study and start to systematically examine the influence of various parameters. This is clearly not possible in full radiation hydrodynamical calculations. Even increasing the size of the setup further while resolving the opacity limit seems computationally affordable. This would be another

step towards sampling the entire IMF, but such a calculation would require additional physics to model the feedback of massive stars on their environment.

5

Paper II: Presentation of an Efficient Structure Finding Algorithm

The material presented in this chapter has been submitted for publication to the *Computational Astrophysics and Cosmology* journal and is published on the arXiv preprint server as *A. Bleuler, R. Teyssier, S. Carassou and D. Martizzi; "PHEW: a parallel segmentation algorithm for three-dimensional AMR datasets. Application to structure detection in self-gravitating flows", arXiv:1412.0510*

5.1. Abstract

We introduce PHEW (**P**arallel **H**i**E**rarchical **W**atershed), a new segmentation algorithm to detect structures in astrophysical fluid simulations, and its implementation into the adaptive mesh refinement (AMR) code RAMSES. PHEW works on the density field defined on the adaptive mesh, and can thus be used on the gas density or the dark matter density after a projection of the particles onto the grid. The algorithm is based on a "watershed" segmentation of the computational volume into dense regions, followed by a merging of the segmented patches based on the saddle point topology of the density field. PHEW is capable of automatically detecting connected regions above the adopted density threshold, as well as the entire set of substructures within. Our algorithm is fully parallel and uses the MPI library. We describe in great detail the parallel algorithm and perform a scaling experiment which proves the capability of PHEW to run efficiently on massively parallel systems.

5.2. Introduction

Over the last decades, computer simulations have become an indispensable tool for studying the formation of structure on all scales in our universe. The common feature of those simulations is the clustering of matter due to self gravity. This clustering is of fractal nature in the sense that - as long as gravity is the dominant force - aggregations of matter turn out to have internal substructures, which are themselves gravitational bound, and may even contain sub- substructures. A crucial tasks in the analysis of simulations is therefore the identification of overdense regions and, ideally, their entire hierarchy of substructure.

First algorithms to perform this task have been invented in the very early days of computer simulations in Astronomy and Astrophysics. A halo finder based on spherical overdensities (SO) was described already four decades ago by Press & Schechter (1974) who used it to find structure in their simulation of 1000 particles. Subsequently, the SO method has become one of the standard methods for halo finding. It consists in growing spherical regions around density peaks and assigning particles inside the spheres to the respective peak based on physical arguments. The also very popular friends-of-friends (FOF) method was introduced to halo finding by Davis et al. (1985). If two particles are separated by less than a user defined linking length, the particles are assigned to the same group. This results in groups of connected particles, the so-called FOF groups. On top of those two methods, a large variety of algorithms has been built over the last two decades: a recent halo finder comparison paper (Knebe et al., 2013) listed 38 different halo finders. For more detailed information about the halo finders which are on the market today, we refer to the series of papers that has emerged from the halo finding comparison project (Knebe et al., 2011, 2013; Onions et al., 2013; Pujol et al., 2014).

Structure finding is not restricted exclusively to the computational cosmology community. Observers, for example, entered the field when they started to automatically identify clumps in position-position-velocity (PPV) cubes resulting from radio observation of molecular clouds.
Stutzki & Guesten (1990) tried to fit the data by sums of triaxial Gaussian-shaped clumps and Williams et al. (1994) identified structure by contouring the dataset at evenly spaced levels without assuming an a priori shape for the clumps. More recently, Rosolowsky et al. (2008) showed how dendrograms can be used to exploit the hierarchy that naturally arises from contouring a PPV cube at multiple emission levels and used this technique to define substructures in molecular clouds.

With such a large choice of astrophysical structure finding tools at hand, one might ask the question why there needs to be yet another one. The trigger for the development of a new analysis tool was our need for "on-the-fly" structure finding in the astrophysical simulation code (Teyssier, 2002), in order to locate gas and/or dark matter clumps while the simulation is running. As pointed out in Knebe et al. (2013) there is a general trend towards "on-the-fly" analysis for many reasons: most modern astrophysical simulations are performed on large computational infrastructure with distributed memory. The sizes of those simulations often exceed the total memory present in commonly used shared memory machines. The structure finding is therefore preferentially performed on the same machine that is running the simulation. Beyond that, the sizes of one single output of such simulations can quickly reach hundreds of GBs, up to several TBs. Storing many outputs for later post-processing is often not possible due to limited disk space, so that keeping only a catalogue of structure is the only viable solution.

Another reason for detecting structures while the simulation is advancing,. is the possibility to couple the results of the halo decomposition to the simulation itself. In Bleuler & Teyssier (2014), for example, we have described a new algorithm for creation of sink particles, based on the properties of gas clumps detected "on-the-fly". This application requires an extremely high frequency at which structure finding must be performed. It must therefore make efficient use of the parallel infrastructure, and deliver good scaling properties for increasing numbers of MPI tasks, up to the number of CPUs the simulation is running on. Otherwise it will unacceptably slow down the simulation.

These requirements resulted in the development of PHEW (**P**arallel **HiE**rarchical **W**atershed), a new structure finding algorithm and its implementation into RAMSES [1]. While PHEW is not based on any pre-existing algorithm, it combines various concepts that have been used in other astrophysical structure finding tools before.

First, PHEW falls into the category of "watershed-based" algorithms. These algorithms assign particles or cells to density peaks by following the steepest gradient, resulting in the so-called "watershed segmentation" (see Section 5.3.1) of the negative density field. Other members of this category are DENMAX (Bertschinger & Gelb, 1991), HOP (Eisenstein & Hut, 1998), SKID (Stadel, 2001), ADAPTAHOP (Aubert et al., 2004), GRASSHOPPER (Potter et al., in prep). Note that in contrast to the aforementioned codes which work on the particles directly, we use a mesh to define the density field[2].

Second, region merging in PHEW is based on the topological properties of saddle surfaces. This is the case as well for HOP, ADAPTAHOP and SUBFIND (Springel et al., 2001). As in the AHF halo finder (Knollmann & Knebe, 2009), PHEW works on the density field deriving from particles that were previously projected onto the AMR mesh. In contrast to AHF, however, we do not use the AMR grid as a way of contouring the density field. A low density region which - for whatever reason - is refined to a high level does not compromise our results. Thus, in the landscape of existing halo finders, PHEW can be seen as filling the gap between P-HOP (Skory et al., 2010) which does not find substructures but is a MPI-parallel version of HOP, and ADAPTAHOP, a multi-threaded software that does find substructures, but has not been yet MPI-parallelized.

The aim of this paper is to present a new structure finding algorithm that: 1- can be applied to any density field defined on an adaptive grid, 2- is capable of detecting substructure, 3- is parallelized using the MPI library on distributed memory systems, and 4- is fast enough to be run at every time step of a simulation without significantly slowing down the calculation. As briefly mentioned above, a previous version of PHEW has already been presented in Bleuler & Teyssier (2014). The algorithm described here differs from the previous one in the sense that it is now fully parallelized. This allows the algorithm to run now efficiently on thousands of CPUs and handle a complex topography with millions of density peaks and a rich hierarchy of substructures.

The article is organised as follows: in Section 5.3 we describe the serial version of the PHEW algorithm. In Section 5.4 we focus on the parallel implementation of the steps presented in Section 5.3. Section 5.5 contains scaling experiments which demonstrate the efficiency of the parallelization. Finally, we summarise and discuss our results, presenting an outlook on possible future work in Section 5.6.

[1] The RAMSES code including PHEW are publicly available and can be downloaded from `http://www.bitbucket.org/rteyssie/ramses`

[2] DENMAX can be considered an in-between case since it uses a uniform grid to compute the density gradient which is then used to directly assign particles to peaks.

5.3. The PHEW algorithm

In this section we describe the serial algorithm. As a starting point, we assume that we have a 3d density field on a AMR grid, particles have been projected onto the grid beforehand. The algorithm can be broken down in four main steps:

- Watershed segmentation
- Saddle point search
- Noise removal
- Substructure merging

In the first step, we assign every cell above a user defined density threshold to a local density maximum by ascending along the steepest gradient. This results in a primary segmentation of the computational volume into "peak patches": regions associated to certain density peak. We establish the connectivity between the peaks by identifying the saddle points. We eliminate the peaks with a low density contrast to the background by merging them to a neighbour through their densest saddle point. The structure surviving the noise removal is considered the finest (sub)-structure. In a last step, we recursively merge the substructure to form larger and larger composite objects.

5.3.1. Watersheds in image processing

Before we start with a more detailed description of the algorithm, we take a quick look over the fence into the field of mathematical morphology and its application to image processing. There, watershed algorithms are a well known and extensively studied tool for image segmentation. The basic idea is that a grayscale image can be thought of as a topographic relief. A drop of water that falls somewhere onto this relief will follow the line of steepest descent until it reaches a local minimum. All points that connect to the same local minimum in that manner form a catchment basin. The watershed algorithm therefore segments the picture into catchment basins. The boundaries of the catchment basins are the actual watersheds. This technique is usually applied to the magnitude of the images gradient. In this way, the watershed lines trace regions of high gradients and segment the original image it into connected regions of small gradients. An excellent overview of the watershed techniques used in image processing is given by Roerdink & Meijster (2000).

When comparing to watershed algorithms used for image segmentation, we have to consider a few aspects where our use case differs from the above one: some watershed algorithms are based on the image pixels being accessed in groups according to their gray level. While an 8 bit image contains only 256 gray levels, our density field is usually represented by an 8 byte float. Looping over all possible gray levels is clearly impossible in our case. Related to that, the limited number of grayscale levels introduces the problem of locally flat regions which do not contain a minimum. Since we use an almost continuous representation of densities we may safely ignore this issue.

Another distinguishing feature of watershed algorithms is whether and how they construct watershed pixels as the boundaries between adjacent catchment basins. This is not important for us as we want to assign every boundary cell to one or the other catchment basin. In this philosophy, the cell surfaces are the actual watersheds which are constructed from the segmented density field after the actual watershed algorithm has finished.

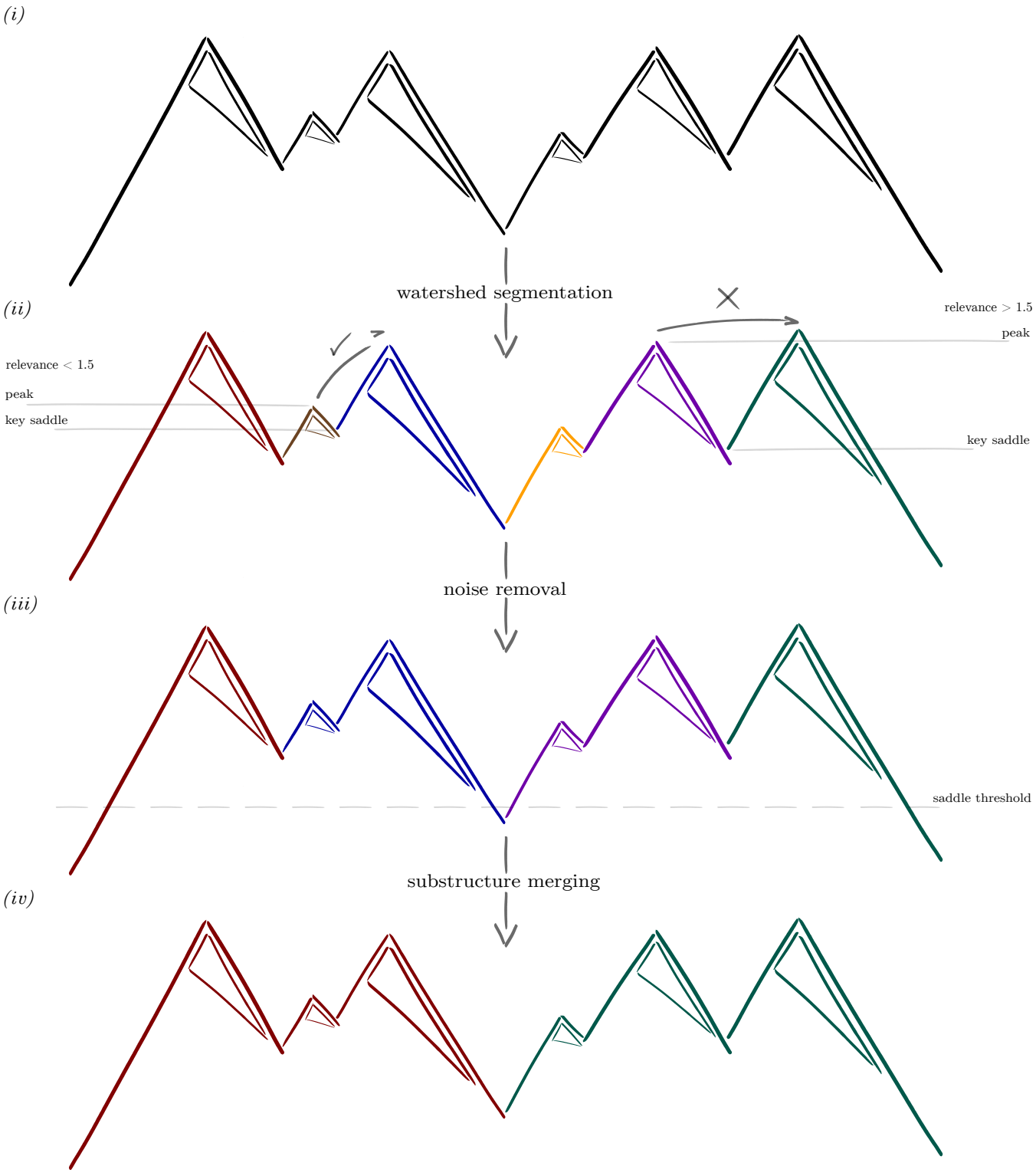

(i)
watershed segmentation
(ii)
relevance < 1.5
peak
key saddle
relevance > 1.5
peak
key saddle
noise removal
(iii)
saddle threshold
substructure merging
(iv)

Figure 5.1

Figure 5.1: Visualization of the main steps of PHEW on a 1d density field (first panel). The segmentation into peak patches is shown in the second panel. Based on the relevance of a peak (peak-to-saddle ratio) we decide whether a peak represents "noise" or substructure. Irrelevant peaks are merged through their highest saddle points (third panel). The surviving objects are labeled as Level 0 clumps and denote the finest level of substructure. The substructure is merged based on a saddle threshold (third panel) into parent structure (fourth panel).

Another important difference is the cost for checking all neighbours of a cell/pixel. Working in 3 dimensions naturally increases the number of neighbours. Using an AMR grid further increases the number of possible neighbours since one has to consider possible neighbours at the same level as the original cell as well as one level above and below. Most importantly, the data structure in an AMR grid is very different from the one of a flat 2d array. The location of neighbouring cells in memory needs to be constructed before a neighbour can be checked for its density. Our main interest lies therefore in reducing the number of neighbours that have to be accessed. These aspects influence the choice of watershed algorithm for our purpose.

5.3.2. Watershed segmentation

In a first step, all cells above the density threshold are marked. We call those cells "test cells". For every test cell the densest neighbouring cell is identified and stored. If a cell has no denser neighbour, it is a local density peak. The peak obtains a peak ID which is written into the `PPatch` label of the corresponding cell. The test cells are sorted by decreasing density. Once sorted, every cell copies the `PPatch` label from its densest neighbour. The previous sorting ensures that the densest neighbour has been accessed before and has therefore already obtained its `PPatch` label. Thus, every cell is assigned to a peak after this one pass. All cells marked with the same `PPatch` label form a peak patch (see Figure 5.1, second panel). Note that our peak patches correspond to the catchment basins introduced in Section 5.3.1. Since we are working on peaks rather than minima, we introduce this new terminology to avoid the cumbersome notion of an "inverted catchment basin". Note that this procedure is very similar to the *hill climbing* method described in Roerdink & Meijster (2000) which was introduced by Meyer (1994).

5.3.3. Saddle point search

Before we can merge peak patches, we have to establish the connectivity between them. All test cells are checked for neighbouring cells that belong to a different peak patch. If such a neighbouring cell is found, the average density of the starting cell and its neighbour is considered as the density at the common surface of the two bordering peak patches. The maximum density on the connecting surface is the saddle between the two peaks and stored. At the end of this step, each peak has its list of neighbouring peaks together with the corresponding saddle point densities. We denote the maximum saddle point of a peak as the "key saddle" and the corresponding neighbour as "key neighbour".

5.3.4. Noise removal

A known problem of the watershed method is over-segmentation. The presence of a huge number of local minima - for example due to random particle noise or transient gas density fluctuations - causes segmentation into as many catchment basins as there are local minima. Generally speaking, there are two possible strategies to deal with this problem: not creating the over-segmentation in the first place or merging over-segmented regions. Preventing over-segmentation the can be obtained using markers to preselect allowed minima (e.g. Moga & Gabbouj, 1998). This usually requires a human intervention, which in our case is not possible. Another way is to use the so-called hierarchical watershed algorithm[3] (Beucher, 1994). Hierarchical watershed algorithms merge artificial catchment basins to more important ones based on some criteria. What we will describe in the following turns our watershed algorithm into a hierarchical algorithm in the Beucher (1994) sense, where our merging criterion is inspired by the notion of a signal-to-noise ratio.

After having previously identified the saddle points, we classify the peaks based on their contrast to the background. We define the contrast as the ratio of the peak density to the key saddle density and name it "relevance". This is sketched in the second panel of Figure 5.1. Every peak is assigned a `NewPeak` label which is initialized to the peaks own peak ID. The peaks are sorted by decreasing peak density. For each peak, the key saddle is determined from the list of saddle points and the relevance is computed. Peaks with a relevance below a relevance threshold are considered noise[4]. If the peak is relevant, it is not touched. For an irrelevant peak, we check whether its key saddle links it to a denser peak. If this is the case, it will inherit the `NewPeak` label from this key peak. As in the watershed segmentation, the previous sorting makes sure that the `NewPeak` labels can propagate through long chains of connected peaks in just one loop. If a peak is both isolated and irrelevant, it is discarded.

When two peaks merge, their lists of saddle points are merged as well. If both peaks used to have a connection to the same third peak, the maximum of the two saddles is kept.

Now, we iterate the procedure: from the updated lists of saddle points, the key saddles are determined. Peaks are accessed in the order of decreasing peak density and irrelevant peaks are merged. After an iteration without any mergers, all irrelevant peaks have been merged or discarded and the noise removal is finished. Note that the described merging process follows exactly the same principle as the watershed segmentation. We have simply replaced cells by peaks, densest neighbour cells with key neighbours and the `PPatch` label by the `NewPeak` label. We call the structures which survive the noise removal **Level 0 clumps**. They constitute the finest structure (see Figure 5.1, third panel) in our hierarchy.

[3]Note that more modern approaches to region merging in image segmentation use the original image for merging while the watershed is computed on the gradient image (e.g., Peng & Zhang, 2011). Using the watershed on the gradient image results in regions of similar gray values, where the densities inside our peak patches are very inhomogeneous. Approaches to region merging are thus fundamentally different in image processing than they are in our case.

[4]The relevance threshold is a user parameter that can be adapted to the setup. 1.5 is our standard choice for identifying gas clumps in RAMSES simulations. For identifying dark matter haloes, the value can be picked according to the expected number of dark matter particles per cell and the resulting Poisson noise in the density.

5.3.5. Saddle threshold merging

If desired, the remaining peaks and their associated clumps can be merged further to form composite clumps. This happens by exactly repeating the previous merging process with a different merging criterion. We have implemented a density threshold for the key saddle as a criterion. If the key saddle density is above that threshold, a peak is merged to its key neighbour (see Figure 5.1, fourth panel). Another possible criterion is the repeated use of the relevance threshold, this time with a higher value.

5.3.6. A hierarchy of saddle points

We have seen in Section 5.3.4 that saddle points are removed in groups or levels by merging through them. All key saddles which link their peak to a denser one are removed at once. Through the merging, other saddle points become key saddles and the next level of saddle points is removed. By repeating this process, a natural hierarchy of saddle points and clumps is produced. In Figure 5.2 we illustrate the construction of this hierarchy. We start with the Level 0 clumps after the noise removal (no substructure except for noise) and assume that the saddle threshold for merging is below any of the saddles depicted in Figure 5.2. The Level 1 saddle points are identified and used for merging. The resulting objects are Level 1 clumps as they have one level of substructure. In general, a Level n clump is formed through a merger which removes a Level n saddle point and contains n levels of substructure. This produces a very natural hierarchy of saddle points and clumps based on the levels of substructure. Note that the level of a saddle point does not reflect its density. A more traditional way of grouping substructure based on the density of the saddle that connects two substructure objects as it is for example produced by ADAPTAHOP can easily be recovered from this hierarchy.

5.3.7. Merging order

We will see in Section 5.4 that we have to drop the idea of sorting the peaks globally when we parallelize PHEW. This will alter the order in which peaks are merged in an unpredictable way. It is therefore crucial that the PHEW allows the order of mergers to change without causing different results. This not true in general. Yet, as we will show in this section, it is the case when we respect the three merging rules:

i) A peak is only merged to a denser one (upward).

ii) A peak is only merged through its key saddle.

iii) The density of the key saddle or the relevance are used as merging criterion.

The result of the merging procedure is uniquely determined by the set of saddle points that is used for merging. This is a subset of all saddle points. In order to affect the outcome of the merging process, changing the order of mergers therefore has to change the set of used saddle points. Let us consider a peak n connected to its key neighbour m through the key saddle s_{nm} at the very beginning of the merging process. The peak density of m is higher than that of n, $m > n$. There are three possible types of mergers related to n or m that can happen before n is considered for merging. We will show that none of them can change the fate of n.

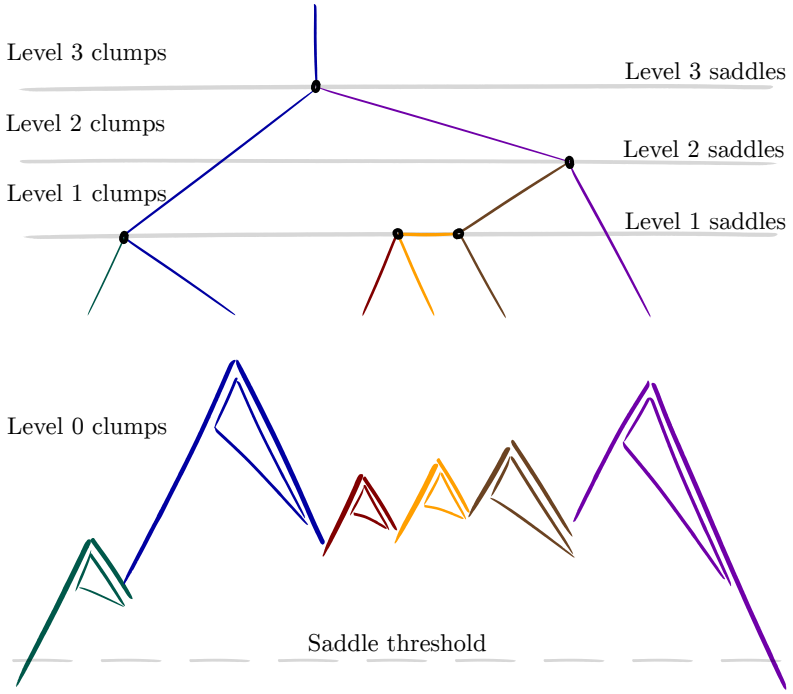


Figure 5.2: Hierarchy of saddle points as it is produced by our merging algorithm. Level n saddle points are used for merging during the n-th round of mergers. Level n clumps emerge from a merger through a Level n saddle point and contain n levels of substructure.

1. A third peak might be merged into m. Due to upward merging, this cannot change the peak density of m and therefore decision if n will be merged into m is not influenced.

2. Peak m might merge into another peak m'. The saddle s_{nm} will still exist, now linking n to m'. Due to upward merging we have $m' > m > n$ which means that n is still the lower of the two peaks connected by $s_{nm'}$. The decision whether n is merged through s_{nm} is unaltered.

3. A third peak i might be merged into n. The peak density of n cannot change due to that since it would mean that peak i had a higher density than n which contradicts the upward merging. The key saddle cannot change because this would mean that peak i had a saddle point s_{ij} higher than s_{nm}. This would imply that the saddle point s_{ni} through which i was merged into n was even higher, $s_{ni} > s_{ij}$ otherwise s_{ni} had not been the key saddle of peak i. Yet, $s_{ni} > s_{ij} > s_{nm}$ contradicts that s_{nm} is the key saddle of peak n. The peak density of n and its key saddle are thus unchanged, therefore the relevance of n is not changed either.

This shows that we can arbitrarily delay the moment when we consider a peak for merging as long as we respect the three merging rules. The mergers happening in the mean time cannot change the properties deciding if and through which saddle this peak will be merged. A possible way to prevent violation of merging rule (ii) is to consider all peaks for merging until no further mergers are possible before any new key saddle of the merged peaks is computed. This results in using the saddle points for merging on a "level-

by-level" basis. This is a key to the parallelization of PHEW since it will allow performing a big number of operations (mergers), in between each round of communication (finding new key saddles). Note that this line of argumentation breaks when we violate merging rule (iii) and use for example the clump mass as merging criterion. The mass is a property that changes with every merger. Therefore, altering the merging order does change the mass of a clump at the moment it is considered for merging and can thus change the decision whether the clumps should be merged or not.

5.4. Parallel implementation

We now turn to the implementation of the previously described steps in a parallel, distributed-memory framework. Where a detailed description of an algorithmic block in words would prevent readability of the paper, we refer the interested reader to a corresponding block written in pseudocode located in Appendix B. We assume that the computational domain has been previously decomposed into non-overlapping spatial domains, each domain containing a partition of the AMR mesh on which the density field is defined. In every MPI task, the local partition of the mesh is referred to as the "active cells". They are wrapped by a thin layer of cells that belong to other tasks. These ghost cells are referred to as belonging to the "virtual boundaries". These virtual boundaries are updated through MPI communication before PHEW is called to make sure that the densities in the virtual boundary cells are equal to the densities in the corresponding active cells hosted by other MPI tasks.

5.4.1. Parallel watershed

The watershed segmentation is *non-local* by nature. This can easily be understood by imagining a mountain ridge. Two drops of water falling onto both sides of the ridge will initially move away into different directions. They might flow into different rivers which flow into different lakes, or they might as well end up in two rivers which join before reaching a lake. The two situations *cannot* be distinguished based on local properties. Parallelization of the watershed algorithm is therefore a non-trivial task. In the literature, one finds various approaches to parallelization for the different watershed algorithms (see e.g. Roerdink & Meijster, 2000). Our technique is very close to the technique described in Moga (1997) and called "hill climbing by locally ordered queues".

Each task performs a loop over all its active cells, in order to identify first the test cells (cells above the density threshold). For faster access, the indices of all test cells are stored in an array. A loop over all test cells is performed where the densities of all neighbouring cells are checked. The index of the densest neighbouring cell is stored for each test cell, since it will be used several times during the algorithm. Note that the densest neighbour of a cell can lie inside the virtual boundary, while test cells are always inside the active domain.

During the first loop, all peaks (local extrema) are counted. After the loop, the number of peaks in each MPI domain are communicated between all MPI tasks, which allows each MPI task to compute a global index (ID) for its peaks (see Figure 5.3). In another loop over test cells, cells which represent a peak are labeled with their global peak ID, all other test cells are initialised with a `PPatch` label equal to zero. The `PPatch` labels are updated inside the virtual boundaries using MPI communication (Figure 5.4, second panel). As explained in Section 5.3, every MPI task computes a permutation which sorts test cells

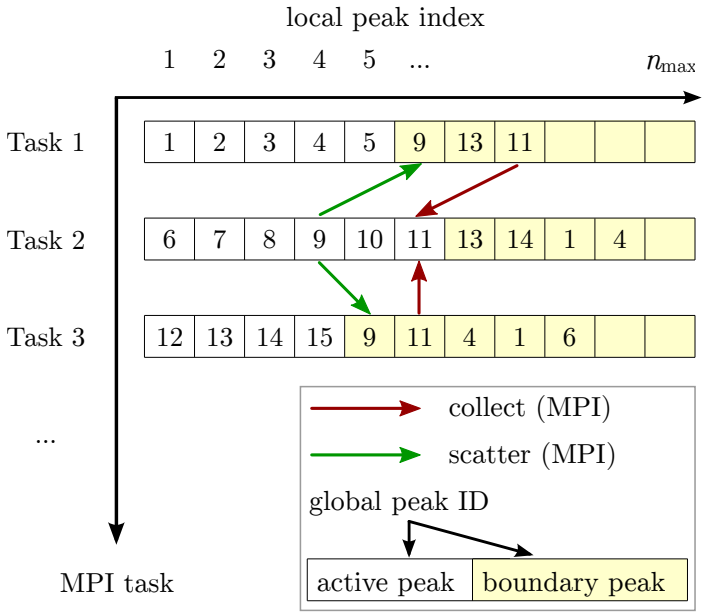


Figure 5.3: Example of peak layout in memory for 3 MPI tasks. The figure shows the global peak ID as a 'function' of the MPI task and the local peak ID. The local peak index for a given global peak ID is stored in a hash table.

in decreasing density order, using the quick sort algorithm (Press et al., 2007). Using this permutation, a sorted loop, where every cell inherits the `PPatch` label from its densest neighbour is performed (Figure 5.4, third panel). During this loop, the number of cells that have changed their `PPatch` label is counted. After the loop, the `PPatch` labels in the virtual boundaries are updated again through MPI communications. This procedure is iterated (Figure 5.4, fourth panel) until no cell inside the entire computational box has changed its `PPatch` label during a full loop. This completes the parallel watershed segmentation.

5.4.2. Virtual peak boundary

As we have already described in Section 5.3, our peak patch merging step is analogous to the segmentation step. The patches now take the role of the cells, the `PPatch` label is replaced by the `NewPeak` label and the densest neighbouring cell is replaced by the key neighbouring patch. As explained before, the parallelization of the peak patch segmentation is exploiting the virtual boundaries surrounding each MPI domain. If we want to use the same strategy to parallelize the merging process, we need the analog of the virtual mesh boundary: a virtual peak boundary. In contrast to our usual virtual mesh boundary, the virtual peak boundary does not represent a fixed region in space. As the merging process advances, new connections appear and new peaks have to be introduced in the virtual peak boundary. Our virtual peak boundary is therefore more dynamic than our virtual mesh boundary.

Figure 5.3 shows a possible layout of peaks in memory. Note the distinction between a peaks global ID and its local index. The latter of the two is the position of the peak in local memory. The peaks that are located inside a tasks MPI domain are called active

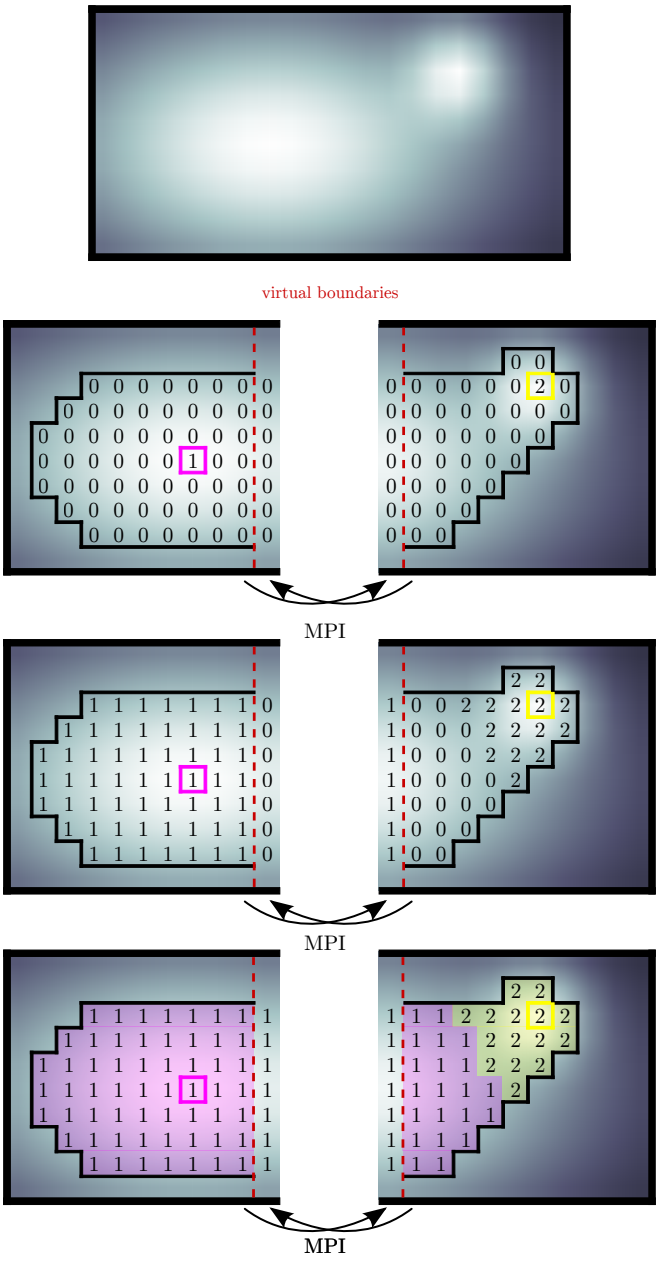


Figure 5.4

Figure 5.4: Parallelization of the watershed segmentation shown on a 2d field. The top panel depicts the computational box with the density field. In the second panel, the two MPI domains and the virtual boundaries are shown, the peaks have obtained their IDs and the cells are labeled. In a loop over all test cells, the `PPatch` labels can propagate inside the MPI domains (third panel). After the loop, the virtual boundaries are updated and the procedure is repeated (fourth panel).

peaks. They take the first N_{active} places in memory. The active peaks are followed by the ghost peaks that belong to the virtual peak boundary. Since it is unknown in advance how much space for ghost peaks will be necessary, we set

$$N_{\text{max}} = \max\{4 \max_{\text{tasks}}\{N_{\text{active}}\}, 1000\}, \tag{5.1}$$

as a default value that can be modified by the user. The preset N_{max} is mostly a large overestimation of the effectively used space in memory for peaks (see, fourth row in Table 5.2), designed to be sufficient for all setups we have tested. However, the memory consumption for peak properties is still negligible compared to the necessary space for the AMR grid. All peak properties such as the peak density are allocated up to N_{max}.

Since every task is aware of its starting number of global peak IDs, switching from global peak ID to local peak index and vice versa is trivial for active peaks. To recover a boundary peaks global ID from its local index, we simply store the global ID in memory at the position of its local index. For the opposite direction we use a hash table that contains the local peak index for a given global peak ID (hash key)[5]. Whenever we introduce a new boundary peak into the virtual peak boundary, it obtains the local peak index corresponding to the first free space in memory. The global peak ID is stored and a hash key is computed. Which peaks need to be present in the virtual peak boundary depends on the connectivity of peaks. The initial state of the virtual boundary will thus be constructed while searching for saddle points that connect the peaks.

5.4.3. The peak communicator

By introducing a peak into the virtual peak boundary, it only obtains a local peak index. No properties except the global peak ID of a newly introduced boundary peak are present at this stage. We now describe how information is transferred from the MPI task which hosts a peak (the "owner" of that peak) into the virtual peak boundaries of other tasks and vice versa. There are two types of communication: inward communication (`collect`, red arrows in Figure 5.3) from all processes which have a certain peak inside their peak boundary to the owner of the peak, and outward communication (`scatter`, green arrows in Figure 5.3) to update the peak properties in the virtual boundaries. When performing a `collect` communication, one has to specify whether one is computing a sum, minimum or maximum of the incoming values belonging to the same peak. When a `scatter` communication is performed, the peak properties of boundary peaks are overwritten with their equivalent from the peaks owner. A typical communication pattern for a peak property is therefore a `collect` communication followed by a `scatter` communication.

Before this communication can effectively be performed, we need to build a communication structure which we refer to as the "peak communicator". We describe the construction of the peak communicator for the situation with 3 MPI tasks depicted in Figure 5.3. We allocate a matrix C of size $N_{\text{task}} \times N_{\text{task}}$. The entry c_{ij} is the number of peaks inside the virtual peak boundary of task i that are owned by task j. Each task builds its line of C in a loop over the boundary peaks by looking at their global peak IDs. Through MPI communication, the lines of C are shared between all task and C is complete. The resulting communication matrix is depicted in Figure 5.5.

[5]We use a simple hash function based on the remainder of a division of the peak ID by a prime number chosen according to the maximum size of the virtual peak boundary. Collisions are dealt with by chaining in the form of a linked list (Knuth, 1998). We found this to be sufficient for our purpose (see Table 5.2).

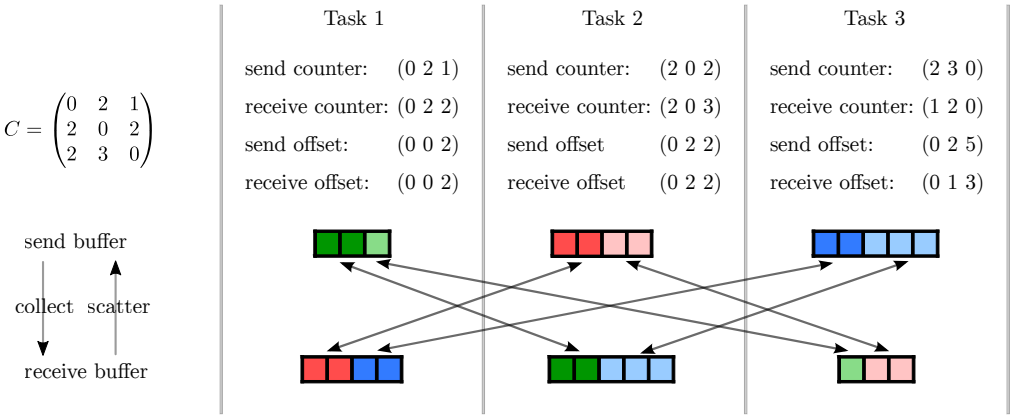


Figure 5.5: Construction of the peak communicator for the example with three MPI tasks shown in Figure 5.3.

We will first consider a `collect` operation. In this case, the lines in C are the send counter vectors and the columns are the receive counter vectors. These counters contain the amount of information that has to be sent/received to/from another MPI task. From those counters, we create the vectors of corresponding offsets. The n-th entry in the send/receive offset vector is simply the sum of the first $n-1$ elements of the send/receive counter vector. The send and receive counters for our example are shown in Figure 5.5. The sum off all entries in the send/receive counters give the length of the send/receive buffer. These buffers are allocated and the send buffer is filled with the peak property to be communicated by looping over all boundary peaks. The receive buffers are filled with the corresponding values through calling MPI_ALLTOALLV[6]. This command efficiently sends each package in the send buffer to the right location in the recipients receive buffers.

In order to complete the setup of the peak communicator, we use the established structure to perform a `collect` communication of the global peak ID. This information allows the identification of a position in the receive buffer with an active peak which completes the communication structure. The peak communicator is rebuilt whenever new peaks have potentially been added to the virtual peak boundary of any MPI task.

The communication structure for the `scatter` communication is identical to the one for the `collect` operation described above. Send and receive buffers, offsets and counters just switch their role.

5.4.4. The saddle point matrix

To keep track of the saddle points, we establish a symmetric saddle matrix M, where the entry m_{ij} is the density of the saddle point connecting the peaks i and j. As most of the peaks patches are not touching each other, we use a sparse matrix representation of M. Note that the indices i, j are the local peak indices, which makes M a sparse matrix of virtual size $N_{\max} \times N_{\max}$. Since we are interested in the maximum entry of each line and the column where it is located when in comes to merging, we keep track of those two values when adding new entries into M. The maximum and its column need to be recomputed

[6] `http://www.mpi-forum.org`

by checking each non-zero element of a line only after values have been removed from the given line in M which reduces the number necessary accesses to the sparse matrix.

The construction of the sparse matrices is performed locally the way described in Section 5.3.3. Whenever a connection is found to a peak that is not yet present in the virtual peak boundary, the given peak is introduced by assigning it a local index. See Algorithm 1 for the pseudocode describing the saddle point search on each task.

5.4.5. Communication of saddle points

We could now use a `collect` communication on the saddle points for every peak in the entire computational box. As a result of that, every task would have access to all saddle points of all his active peaks. The global key saddle and key neighbour could then be determined by every MPI task for his active peaks. However, this approach would introduce a lot of communication and unnecessarily fill the sparse saddle matrices. The only necessary information to perform one iteration in the merging process is the (global) key saddle density of a peak and the corresponding key neighbour. This global maximum saddle can be found by comparing the local maxima of each MPI task. We thus minimise communication by performing a `collect` communication only on the local maximum of each row in the saddle point matrix. Together with the local maximum saddle density, we `collect` the global peak ID that denotes the local key neighbour. The owner of a peak can now compute the global key saddle for a given peak by comparing all the local maxima. The global peak ID that was received from the MPI task which hosts the global key saddle is the key neighbour of the peak. If not already present, the key neighbour is introduced into the virtual peak boundary of the owner task and the key saddle density is written into the sparse saddle matrix of the owner. Every MPI task can now perform a complete iteration in the merging process without any further communication of saddle point densities.

5.4.6. Merging in parallel

We are now set for the actual merging of the peaks. We introduce two new peak properties: a logical variable called `alive` which is initialised to `true` and set to `false` when a peak is merged into another one, and the `NewPeak` label which is initialised to the global peak ID for all active peaks. These two new properties and the peak density are updated in the virtual peak boundaries using a `scatter` communication. A permutation which sorts the active peaks by decreasing density is computed. Now we propagate the `NewPeak` label through the key saddles in a level-by-level fashion. On each level, we iterate until no `NewPeak` label is moved, while the virtual boundaries are updated after every iteration. This is perfectly analogous to the parallel watershed segmentation. After every level of saddle points we update the `alive` variable, the saddle point matrices and the virtual boundaries. The merger routine is described in Algorithm 2 in pseudocode. The substructure merging is performed in exactly the same way, we just replace the relevance threshold by the saddle density threshold.

5.5. Scaling test

We use a previously run cosmological dark matter simulation with 512^3 particles for a scaling experiment. We restart the simulation from the output corresponding to redshift $z = 0$ using various numbers of MPI tasks. Before PHEW can run, we project the particle

density onto the AMR grid using the CIC (Cloud-In-Cell, Hockney & Eastwood, 1981) algorithm. Once we have constructed the grid-based density field, we run PHEW with a density threshold of 80 times the cosmological critical density (noted ρ_{crit}) and a relevance threshold of 3. After merging the peak patches into Level 0 clumps (sub-haloes), we merge to form haloes by applying a saddle threshold of $200\rho_{\mathrm{crit}}$. The first column in Table 5.1 summarizes parameters and runtime statistics obtained for 1024 tasks. We see a rich hierarchy of saddle points spread over many levels. The numbers of iterations necessary show that there is structure extending over several domain boundaries at every stage of the process (peak patches, clumps, haloes). Note that we PHEW finds *exactly* the same structures, independent of the number of MPI tasks that have been used. This empirically confirms what we have described in Section 5.3.7. It is also worth mentioning that the iteration pattern looks surprisingly similar for the other 512^3 runs in our scaling experiment. The total number of necessary iterations increases from 35 to 45 when going from 32 to 2048 tasks while it would be only 3 when for the serial algorithm. An example of the hierarchical structure that is found by PHEW is shown in Figure 5.8 which depicts a halo with four levels of substructure taken from our scaling experiment.

Table 5.1: Parameters and some runtime statistics for the 1024 task runs of the experiment.

N_{parts}	512^3	1024^3
N_{tasks}	1024	1024
Density threshold	$80\,\rho_{\text{crit}}$	$80\,\rho_{\text{crit}}$
Relevance threshold	3	3
Saddle threshold	$200\,\rho_{\text{crit}}$	$200\,\rho_{\text{crit}}$
Number of test cells	104 360 968	835 609 288
Number of density peaks	6 714 764	53 994 995
Number of relevant clumps	1 311 208	10 612 079
Number of haloes[7]	521 185	4 234 746
Runtime	8.0 s	38.9 s
Number of iterations for...		
...watershed segmentation	7	9
...noise removal		
Level 1	7	7
Level 2	5	6
Level 3	4	4
Level 4	2	3
Level 5	1	2
Level 6	1	1
Level 7	1	1
Level 8		1
...substructure merging		
Level 1	4	3
Level 2	3	4
Level 3	3	3
Level 4	2	2
Level 5	1	2
Level 6	1	1
Level 7		1

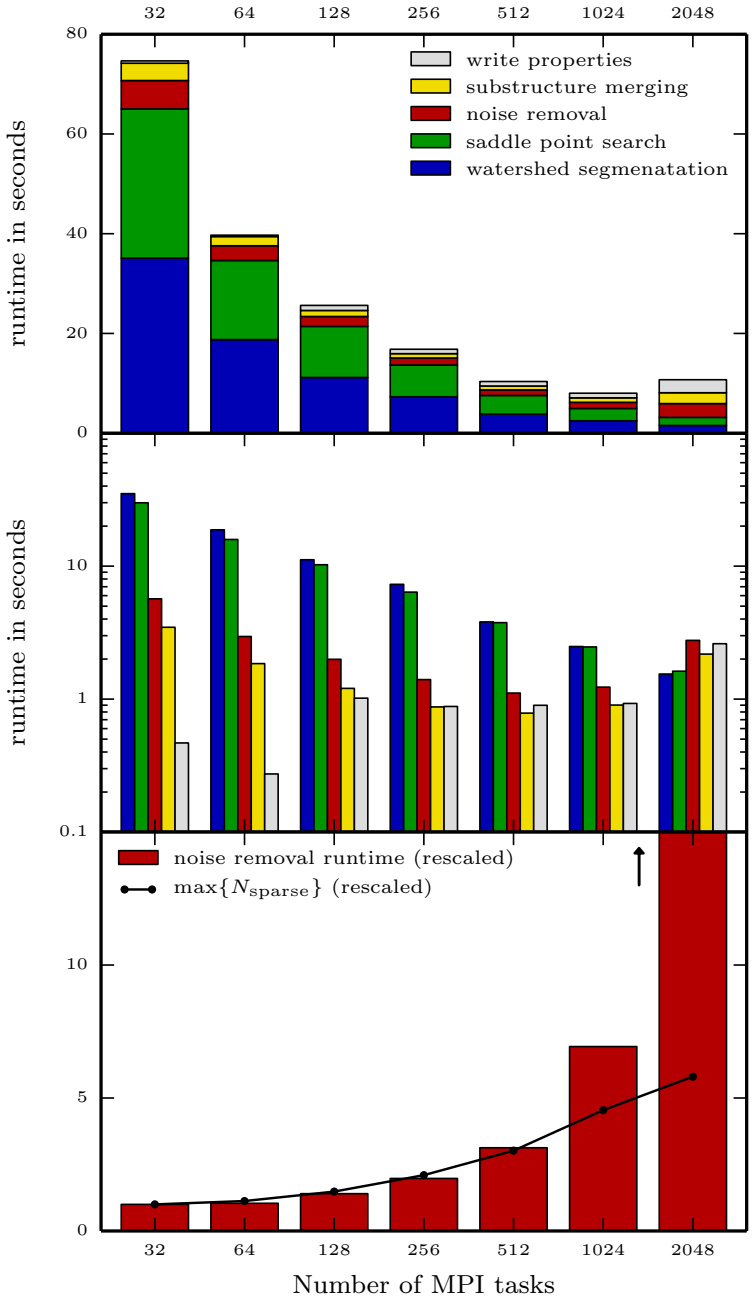


Figure 5.6: Scaling properties of the different parts in PHEW obtained by restarting a cosmological dark matter simulation with 512^3 particles at redshift $z = 0$. The top two panels show the runtimes of the different algorithmic blocks in PHEW. The peak patch segmentation and the saddle point search exhibit excellent scaling in the entire range of MPI tasks that we have tested. The merging in our test scales well up to ~ 256 MPI tasks. The bottom panel shows the maximum number of sparse matrix elements over all MPI tasks compared to $1/N_{\mathrm{tasks}}$ and rescaled to one at 32 MPI tasks. The increase seen in this number for of tasks is due to the growing load imbalance in terms of peaks per task and the increase in the surface to volume ratio of the domain segmentation. It explains the increase of the scaled runtime of the noise removal very well up to 512 tasks. The overall scaling of the algorithm is satisfactory up to 1024 MPI tasks which is four times the number of CPUs the original simulation was run on.

Table 5.2: Runtime diagnostics for the parallelization of PHEW when various numbers of MPI tasks are used. N_{active} and N_{ghost} are the number of active peaks and ghost peaks respectively and $N_{\text{tot}} = N_{\text{active}} + N_{\text{ghost}}$ denotes the total number of peaks per MPI task. N_{sparse} is the number of entries in the sparse saddle matrix and $N_{\text{collisions}}$ gives the number of hash table collisions. Sums, maxima and averages are taken over the all MPI tasks.

N_{tasks}	32	64	128	256	512	1024	2048
Load imbalance $\left(\frac{\max\{N_{\text{tot}}\}}{\text{avg}\{N_{\text{tot}}\}}\right)$	1.4	1.5	1.8	2.4	2.8	3.3	3.9
Surface effect $\left(\frac{\sum N_{\text{ghost}}}{\sum N_{\text{active}}}\right)$	0.0087	0.012	0.016	0.021	0.030	0.040	0.055
Connectivity $\left(\frac{\sum N_{\text{sparse}}}{\sum N_{\text{tot}}}\right)$	9.4	9.4	9.4	9.3	9.3	9.3	9.2
$\max\left\{\frac{N_{\text{ghost}}}{N_{\text{active}}}\right\}$	0.012	0.017	0.044	0.064	0.10	0.15	0.24
$\max\{N_{\text{tot}}\}$	3.0×10^5	1.6×10^5	9.6×10^4	6.4×10^4	3.8×10^4	2.2×10^4	1.3×10^4
$\max\{N_{\text{sparse}}\}$	3.3×10^6	1.8×10^6	1.2×10^6	8.7×10^5	6.3×10^5	4.7×10^5	3.0×10^5
$\max\{N_{\text{collisions}}\}$	4	3	2	3	16	17	13

In our numerical experiment, PHEW was run five times in a row, for five main simulation time steps following the restart. We measure the total runtime of each call to PHEW as well as the time spent on the different algorithmic steps. We find the variance of the runtimes to be negligible and conclude that the timings are stable. Note that the preliminary construction of the density field is performed inside the watershed segmentation block. However, the CIC algorithm is quick compared to the watershed segmentation. We also measure the amount of time necessary for each MPI task to write the properties of the structure inside its domain to disk.

The runtimes for the various numbers of MPI tasks are plotted in the top two panels of Figure 5.6. The top panel shows an excellent scaling of the algorithm up to 1024 MPI tasks which is four times the numbers of tasks that were used to perform the original simulation. In this regime, the total runtime of PHEW is dominated by the watershed segmentation and the saddle point search. The most costly operations inside those two blocks are the construction and access of neighbouring cells. The total workload of those blocks thus scales linearly with the number of test cells per MPI task.

The second panel shows that the runtime of those two blocks does actually scale over the entire range of numbers of tasks that we have tested. The second panel in Figure 5.6 shows that the merging procedures scale well up to 256 tasks. The scaling of the merging process in this region is mainly controlled by two effects: with a growing number of tasks, the load imbalance of the peaks between the different MPI tasks increases. This is unavoidable as the domain decomposition is optimised for all AMR cells, not for the test cells only, and even less for the peak patches. The second reason is the growing ratio of surface to volume as the computational box is divided in smaller parts. This results in more ghost peaks per active peak which causes a higher workload per active peak. Those two effects are quantified in the first two rows of Table 5.2.

The solid line in the bottom panel of Figure 5.6 is a result of both effects mentioned above. It depicts $\max\{N_{\mathrm{sparse}}\}$, the maximum number of used sparse matrix elements over all MPI tasks. In perfect scaling conditions, this number would decrease as $1/N_{\mathrm{tasks}}$. We thus multiply $\max\{N_{\mathrm{sparse}}\}$ by N_{tasks} and rescale to one at 32 tasks. We compare this to the runtime of the noise removal (also scaled). We observe that this "worst case" number of entries in the sparse saddle point matrix does explain the scaling of the merging process up to 512 tasks. Beyond that, we believe that MPI communications become the performance bottleneck.

In Table 5.2 we also show the maximum ratio of ghost peaks to active peaks. For 2048 tasks we have a value of 24%. This shows that the number N_{max} defined in Equation 5.1 is an overestimation of the effectively used memory for ghost peaks for this setup. In the same table, we also list the number of hast table collisions. There are very few collisions as the hash table is far from filling up and we conclude that the relatively simple hash function that we use is good enough for our purpose. Another fact worth mentioning is the relatively constant ratio of non-zero entries in saddle point matrix to the number of peaks seen in the third line of Table 5.2. Divided by two (due to the symmetry of the saddle point matrix), this number gives a good idea of the effective number of neighbours per peak.

As a second test we perform a "weak scaling" comparison of our 1024 task run with another 1024 task run but this time on a larger, 1024^3 particle box. The second column of Table 5.1 lists the statistics of that run. The numbers of test cells, peaks, clumps and haloes all increase by the expected factor of ≈ 8. We thus divide the runtimes of PHEW

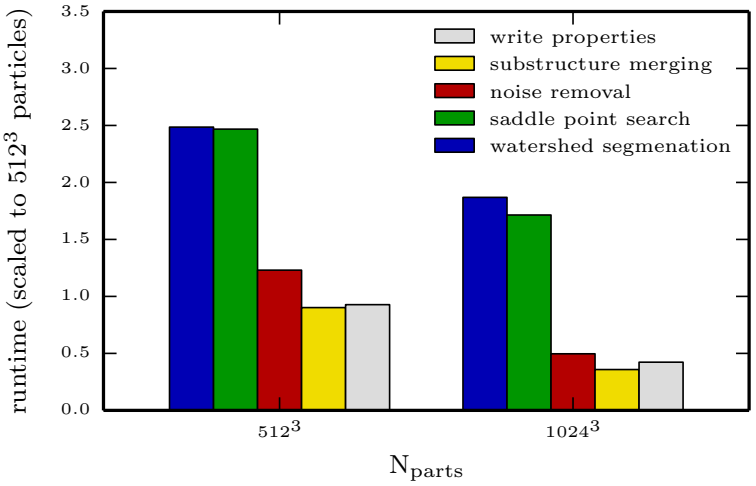


Figure 5.7: Weak scaling comparison of PHEW using 1024 tasks to find structure in a 512^3 and a 1024^3 particle cosmological box. The PHEW runtimes for the 1024^3 box are divided by a factor of 8 for comparison with the runtimes for the 512^3 box. Increasing the size of the dataset improves the scaling of PHEW for large numbers of MPI tasks.

for this setup by 8 and compare to the runtime of the 1024 task run on the 512^3 box. This comparison is plotted in Figure 5.7. The figure shows that the runtime per data decreases for all parts of PHEW by increasing the size of the data. Especially the efficiency of merging routines benefit a lot from the increased size of the dataset. We thus conclude that we can enlarge the range of N_{task} where PHEW scales well, by increasing the size of the simulation.

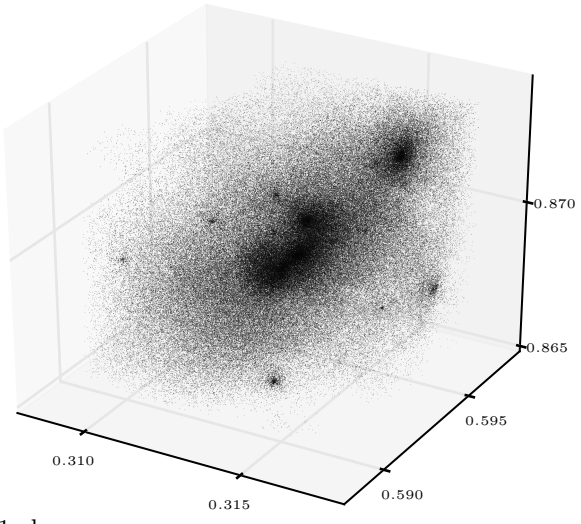
5.6. Conclusions

We have presented PHEW, a new structure finding algorithm and its MPI parallel implementation into the AMR code RAMSES. PHEW finds density peaks and their associated regions in a 3D density field by performing a watershed segmentation. The merging is based on the saddle point topology. We have described a two-step approach to merging. In a first step, we merge irrelevant density fluctuations which we consider as noise. In a second step we merge the finest substructure hierarchically, into large, connected regions above the adopted density threshold. This merging process naturally results in a tree-like representation of substructure similar to the dendrograms presented by Rosolowsky et al. (2008).

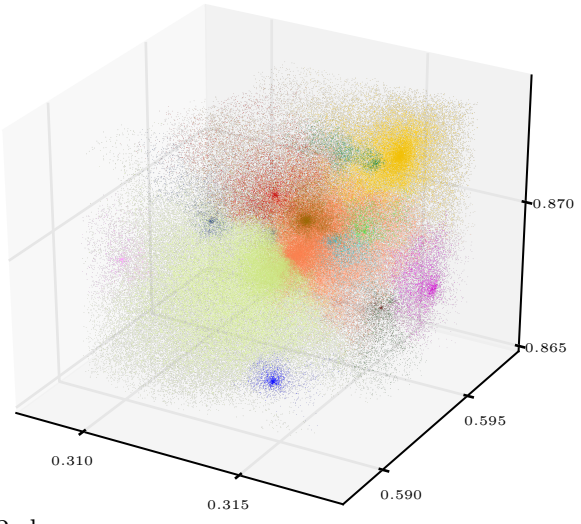
The main focus of this article is on the parallel implementation of the algorithm which we have described in detail. Our implementation is truly parallel, meaning that it produces *exactly* the same results for varying numbers of MPI tasks. To test the parallelization of PHEW, we have performed a scaling experiment on a snapshot from a cosmological dark matter simulation. We have found excellent scaling in the relevant range of MPI tasks. When using the same number of MPI tasks that was used for the actual simulation, the runtime of PHEW $\sim 10\%$ the time it takes to advance the simulation by one time step. This allows for frequent usage of PHEW on-the-fly and thus more fine-grained information about how matter assembles in simulations.

PHEW has similarities with already existing watershed based halo finders, such as DENMAX (Bertschinger & Gelb, 1991), HOP (Eisenstein & Hut, 1998), SKID (Stadel, 2001),

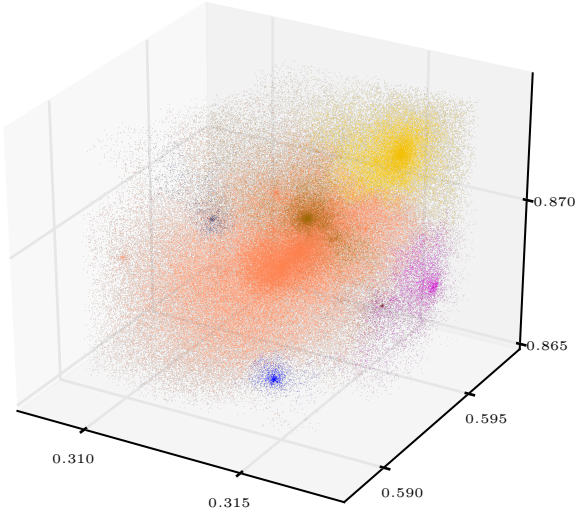
Dark matter particles



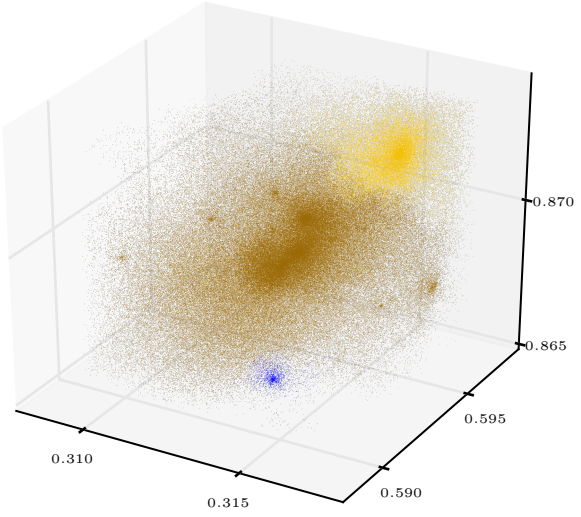
Level 0 clumps



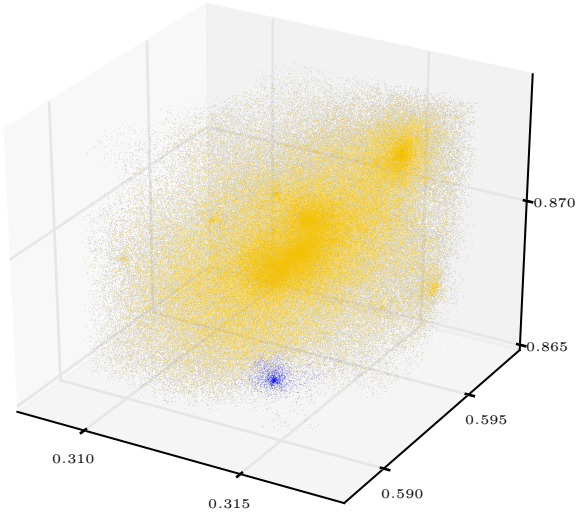
Level 1 clumps



Level 2 clumps



Level 3 clumps



Main halo

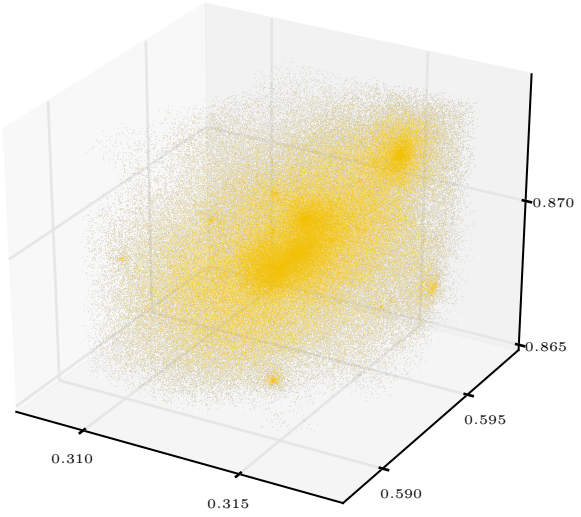


Figure 5.8

Figure 5.8: Visualization of PHEW applied to a dark matter halo. We show a small sub-volume of the 512^3 particle box used in our scaling experiment. The coordinates indicate the fraction of the box size. The sub-volume contains $\approx 2 \times 10^6$ particles. The objects that emerge after the noise removal (Level 0 clumps) are indicated in the second panel, where all particles belonging to the same object share a color. Every subsequent panel shows the status after a further round of merging as it is described in Section 5.3.6.

ADAPTAHOP (Aubert et al., 2004), GRASSHOPPER (Potter et al., in prep), but these are either not yet parallelized, do not find substructure or work only on particles. On a first sight, it looks like our approaches to define substructure or parallelization cannot be applied to particle-based data structures since we operate on a mesh-defined density field, while the other codes work on the particle distribution directly. However, the only two concepts that we use which are naturally provided by the grid, namely a local density and the notion of a neighbour, can be also defined for other data structures that do not rely on a grid. Once these properties are defined, the algorithm presented in this paper can be applied to particle data in the same way as we apply it to grid data.

At the current stage, our implementation of PHEW is a topological tool only, meaning that it identifies regions in space disregarding physical properties such as the kinetic or gravitational energy of the matter in that volume. For the application of PHEW as a genuine halo finder, we need to develop an unbinding procedure, which removes dark matter particles from regions they are not gravitationally bound to. We will exploit our hierarchical decomposition into substructure, to pass unbound particle to larger and larger regions, until the particles remain bound. This will unambiguously define the parent halo (or sub-halo) of the particles.

5.7. Acknowledgements

We want to thank Stephane Colombi for his advice on substructure merging. Furthermore we thank Doug Potter for helpful discussions about programming techniques. The computations leading to this publication have been performed at on the zBox4 and Schroedinger Supercomputers at the University of Zurich and at the Swiss Supercomputing Centre CSCS in Lugano. This work has been supported by the Swiss National Science Foundation SNF under the project "Computational Astrophysics" and the PASC co-design project "Particles and Fields".

5.8. Appendix A: Glossary

Clump:

We use the word clump for the structure after the noise removal. It is the smallest structure that is not considered noise.

Key saddle:

The highest saddle point connecting a peak to any neighbouring peak is considered the key saddle. Note that this definition slightly deviates from the one traditionally used in topography.

Key neighbour:

A peaks key neighbour is the peak it is connected to through the key saddle.

Neighbouring cell:

Every cell with a common face, edge or corner is considered a neighbour to a given cell.

Neighbouring peak:

If a cell inside peak patch i is neighbouring a cell in peak patch j, their peaks are considered neighbouring peaks.

Noise:

A peak with a small relevance (usually less than 1.5) is considered noise.

Owner:

We denote the MPI task where a given peak is active as the owner of that peak.

Peak:

We denote every cell hosting a local density maximum as a peak.

Peak patch:

Every cell is unambiguously connected to one single density peak by recursively assigning it to the densest neighbouring cell. All cells belonging to a certain peak form the so-called peak patch. The peak patch is the equivalent to the watershed catchment basins for the negative density field.

Relevance:

The relevance is defined as the ratio of a peaks density to its key saddle density or the density threshold in case of an isolated peak patch. This term is closely related to the topographical term "prominence", which denotes the altitude difference of a peak to its highest saddle which connects the peak to a higher neighbour.

Saddle point:

The density maximum on the connecting surface between two peak patches is located at the saddle point connecting the two peaks.

Test cell:

Cells with a density above the adopted density threshold are called test cells. Only those are considered in our analysis.

5.9. Appendix B: Algorithmic blocks in pseudocode

```
1  for  testcell ∈ {testcells} do
2  |    for neighbour ∈ {neighbours} do
3  |    |    if (PPatch [neighbour] ≠PPatch [testcell]) and (PPatch [neighbour] > 0) then
4  |    |    |    i=GetLocalPeakIndex (PPatch [testcell])
5  |    |    |    j=GetLocalPeakIndex (PPatch [neighbour])
6  |    |    |    if AverageDensity (testcell,neighbour) > SaddleMatrix [i,j] then
7  |    |    |    |    SaddleMatrix [i,j]=AverageDensity (testcell,neighbour)
8  |    |    |    |    SaddleMatrix [j,i]=AverageDensity (testcell,neighbour)
9  |    |    |    end
10 |    |    end
11 |    end
12 end
```

Algorithm 1: Pseudocode describing the construction of the local saddle point matrices.

```
1  Preparatory step - initialize two peak-based properties.
2  for peak ∈ {active peaks} do
3  |  alive [peak]=1
4  |  NewPeak [peak]=GlobalPeakID [peak]
5  end

6  Loop over Levels in the saddle point hierarchy.
7  mergers=1
8  while mergers > 0 do
9  |  mergers=0

10 |  Propagate the NewPeak label through key saddle points.
11 |  LevelMergers=1
12 |  while LevelMergers > 0 do
13 |  |  LevelMergers=0
14 |  |  CommunicateSaddlepoints
15 |  |  BuildPeakCommunicator
16 |  |  ScatterCommunicate (PeakDensity,NewPeak,alive)
17 |  |  for peak ∈ {sorted active peaks} do
18 |  |  |  if alive [peak]> 0 then
19 |  |  |  |  PSratio=PeakDensity [peak]/KeySaddle [peak]
20 |  |  |  |  if PSratio > 1.5 and PeakDensity [KeyNeighbor [peak] ]> PeakDensity [peak] then
21 |  |  |  |  |  NewPeak [peak]=NewPeak [KeyNeighbor [peak]]
22 |  |  |  |  |  LevelMergers=LevelMergers+1
23 |  |  |  |  end
24 |  |  |  end
25 |  |  end
26 |  |  ScatterCommunicate (NewPeak)
27 |  |  LevelMergers=MPIsum (LevelMergers)
28 |  end

29 |  For every merger, merge the corresponding lines in the saddle point array.
30 |  for peak ∈ {all peaks} do
31 |  |  if GlobalPeakID [peak] ≠ NewPeak [peak] then
32 |  |  |  NewIndex=GetLocalPeakIndex (NewPeak [peak])
33 |  |  |  for column ∈ {matrix columns} do
34 |  |  |  |  if SaddleMatrix [peak,column] > SaddleMatrix [NewIndex,column] then
35 |  |  |  |  |  SaddleMatrix [NewIndex,column]=SaddleMatrix [peak,column]
36 |  |  |  |  |  SaddleMatrix [column,NewIndex]=SaddleMatrix [peak,column]
37 |  |  |  |  end
38 |  |  |  end
39 |  |  |  SaddleMatrix [NewIndex,peak]=0
40 |  |  |  SaddleMatrix [NewIndex,NewIndex]=0
41 |  |  end
42 |  end
43 |  BuildPeakCommunicator

44 |  Set alive to zero for dead peaks and count mergers.
45 |  for peak ∈ {active peaks} do
46 |  |  if GlobalPeakID[peak] ≠ NewPeak [peak] and alive [peak]==1 then
47 |  |  |  alive [peak]=0
48 |  |  |  mergers=mergers+1
49 |  |  end
50 |  end
51 |  ScatterCommunicate (alive)
52 |  mergers=MPIsum (mergers)

53 |  Remove saddle points linking to dead peaks.
54 |  for peak ∈ {all peaks} do
55 |  |  for column ∈ {matrix columns} do
56 |  |  |  if alive [peak]==0 or alive [column]==0 then
57 |  |  |  |  SaddleMatrix [peak,column]=0
58 |  |  |  end
59 |  |  end
60 |  end
61 end
```

Algorithm 2: Pseudocode describing the parallel merger procedure.

6

PROSPECTS

I finalize my manuscript by presenting an outlook on work that could be done in the future. The final lines of a doctoral thesis are not the place to be overly modest. Consequently, I focus on big goals, although some of them might eventually turn out to be too challenging. I am happy that I am given the opportunity to continue my work here at the Institute for Computational Science. I will therefore have the chance to tackle at least some of the problems mentioned here myself.

6.1. More mass

In Chapter 4 we have described the simulation of star formation in a $1000\,\mathrm{M}_\odot$ molecular cloud. Our model is optimized for computational speed which allows us to run such a simulation in comparably little time. Evolving a $1000\,\mathrm{M}_\odot$ molecular cloud until 10% of its initial mass have been converted into stars is sufficient to sample the IMF up to stellar masses $\lesssim 10\,\mathrm{M}_\odot$, but we can not expect to fully enter the regime of massive star formation. Additionally, we are still below the typical mass range of the giant molecular clouds observed in our galaxy, $10^4 - 10^6\,\mathrm{M}_\odot$. In a next step we should therefore increase the size of our setup even further. Simulations of molecular clouds with a mass $> 10^3\,\mathrm{M}_\odot$ have been performed before (e.g., Padoan et al., 2014; Smith et al., 2009), but to our knowledge not while still resolving the opacity limit.

In our case, going from $10^3\,\mathrm{M}_\odot$ to $10^4\,\mathrm{M}_\odot$ will not be as easy as just initializing a bigger cloud and waiting for 10 weeks instead of only 1 week. In such a simulation we would expect to form stars with a mass larger than $10\,\mathrm{M}_\odot$ which exert other types of feedback on the surrounding gas than just the discussed heating through infrared radiation. At least the effects of radiation pressure (Skinner & Ostriker, in prep.) and ionizing radiation (e.g., Peters et al., 2011; Walch et al., 2013) and maybe outflows (e.g., Wang et al., 2010) would have to be modeled in some computationally affordable way. Again, such models would have to be challenged by comparing them with physically more complete models using smaller or not as highly resolved setups. The additional feedback from massive stars might eventually be a way to self-consistently terminate the accretion process and disperse the

molecular cloud rather than just stopping the simulation at some star formation efficiency. This would be an genuine breakthrough because it would shed light one of the big problems of star formation: the low observed star formation rate (Krumholz, 2014).

6.2. More physics

Magnetic fields, stellar outflows, ionizing radiation and radiation pressure - the list of physical effects that we have neglected so far is long. Many of the numerical schemes necessary to add the additional physics are implemented in RAMSES and ready to use (MHD: Fromang et al. (2006), RAMSES-RT: Rosdahl et al. (2013)). Other effects like stellar outflows would still need to be added to obtain a physically more complete scheme. As we have demonstrated for the case of RAMSES-RT, this can significantly raise the computational costs of the simulations, but not necessarily to the point where they become completely unfeasible. A very rough estimate based on the simulations performed in Chapter 4 would suggest that the $1000\,\mathrm{M}_\odot$ setup from Section 4.4 would require $10^6 - 10^7$ CPU hours to run when using RAMSES-RT. This would definitely be expensive simulation to run, but not impossible.

Alternatively one could burn a lot of "brain hours" rather than CPU hours and try to design faster computational methods to treat the additional physics. A contemporary trend in the field of high performance computing is the use of graphics cards (GPUs) as accelerators that can very efficiently perform a vast amount of operations in parallel. The option of performing radiative transfer on GPUs in the RAMSES framework has already been explored by Aubert & Teyssier (2010) as a post-processing step and, more recently, by Ocvirk (2014) for cartesian grids but in coupling with the hydrodynamical evolution.

6.3. More resolution

At the other side of the spectrum, one can give up on a sampling of the IMF in one simulation and focus on the collapse of one single core at very high resolution and with accurate physics. A motivation for turning the attention to smaller scales is given by the fact that formation of cores is arguably better understood than the subsequent formation of individual stars. What sounds like a shift of paradigm compared to the work presented in this thesis connects to the discussion of numerical convergence - or its absence - when sink particles are used (see Section 3.7.1.2): currently, we introduce sink particles at the entry of the first Larson core phase. It is not surprising that approximating such a slowly contracting region by a point mass at a rather arbitrary moment during the contraction does not produce converged results. The point where the gas enters the phase of the second collapse from AU scale down to the actual radius of the protostar could be fairly well approximated by a point-like singularity and thus represents the natural point for the introduction of sink particles. In such a setup it might even be possible to demonstrate numerical convergence in the presence of sink particles. This would add a lot of credibility to the simulation results.

Previous simulations of fragmenting cores have come close to the necessary resolution for the onset of the second collapse. Stamatellos et al. (2011) for example introduce sink particles at a density of $10^{-9}\,\mathrm{g\,cm^{-3}}$. However, I do not know of any calculation of a fragmenting core that has moved beyond that into the second collapse.

6.4. More analysis

So far we have focused on the properties of the sink particles when analyzing our simulation results. These properties are well defined, easy to extract from the simulation and - by identifying sinks as stars - can be compared to observational results in a straightforward manner. What we have ignored to a large extent are the properties of the remaining gas. Here lies a wealth of possibilities for further analysis of simulation and comparison to observations. Some studied performed in recent years (Commerçon et al., 2012; Peters et al., 2010) have led the path and it is very likely that the fabrication of "synthetic observations" from simulations will become more standard in the near future. This is likely to yield many "observables" which are not reproduced accurately by simulations, but it will sharpen our understanding of the predictive power of numerical simulations in the field of star formation.

Bibliography

Aarseth S. J., 1963, Monthly Notices of the Royal Astronomical Society, 126, 223

Agertz O. et al., 2007, Monthly Notices of the Royal Astronomical Society, 380, 963

André P. et al., 2010, Astronomy and Astrophysics, 518, L102

Aubert D., Pichon C., Colombi S., 2004, Monthly Notices of the Royal Astronomical Society, 352, 376

Aubert D., Teyssier R., 2010, The Astrophysical Journal, 724, 244

Bastian N., Covey K. R., Meyer M. R., 2010, Annual Review of Astronomy and Astrophysics, 48, 339

Bate M. R., 2009, Monthly Notices of the Royal Astronomical Society, 392, 590

Bate M. R., 2012, Monthly Notices of the Royal Astronomical Society, 419, 3115

Bate M. R., Bonnell I. A., Price N. M., 1995, Monthly Notices of the Royal Astronomical Society, 277, 362

Bate M. R., Burkert A., 1997, Monthly Notices of the Royal Astronomical Society, 288, 1060

Berger M., Colella P., 1989a, Journal of computational Physics, 82, 64, lawrence Livermore Laboratory Report No. UCRL-97196.

Berger M. J., Colella P., 1989b, Journal of computational Physics, 82, 64

Bergin E. A., Tafalla M., 2007, Annual Review of Astronomy and Astrophysics, 45, 339

Bertschinger E., Gelb J. M., 1991, Computers in Physics, 5, 164

Beucher S., 1994, in Mathematical Morphology And Its Applications To Image Processing, Serra J., Soille P., eds.

Binney J., Tremaine S., 2011, Galactic dynamics. Princeton university press

Bleuler A., Teyssier R., 2014, Monthly Notices of the Royal Astronomical Society, 445, 4015

Bond J., Cole S., Efstathiou G., Kaiser N., 1991, The Astrophysical Journal, 379, 440

Bondi H., 1952, Monthly Notices of the Royal Astronomical Society, 112, 195

Bonnell I., Larson R., Zinnecker H., 2007, Protostars and planets v, ed

Bonnell I. A., Vine S. G., Bate M. R., 2004, Monthly Notices of the Royal Astronomical Society, 349, 735

Bonnor W. B., 1958, Monthly Notices of the Royal Astronomical Society, 118, 523

Boss A. P., Bodenheimer P., 1979, The Astrophysical Journal, 234, 289

Chabrier G., 2003, Publications of the Astronomical Society of the Pacific, 115, 763

Chabrier G., 2005, in Astrophysics and Space Science Library, Vol. 327, The Initial Mass Function 50 Years Later, Corbelli E., Palla F., Zinnecker H., eds., p. 41

Collins D. C., Kritsuk A. G., Padoan P., Li H., Xu H., Ustyugov S. D., Norman M. L., 2012, The Astrophysical Journal, 750, 13

Commerçon B., Audit E., Chabrier G., Chièze J.-P., 2011a, Astronomy and Astrophysics, 530, A13

Commerçon B., Hennebelle P., Audit E., Chabrier G., Teyssier R., 2008, Astronomy and Astrophysics, 482, 371

Commerçon B., Hennebelle P., Henning T., 2011b, The Astrophysical Journal Letters, 742, L9

Commerçon B., Hennebelle P., Audit E., Chabrier G., Teyssier R., 2010, Astronomy and Astrophysics, 510, L3

Commerçon B., Launhardt R., Dullemond C., Henning T., 2012, Astronomy and Astrophysics, 545, 98

Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, The Astrophysical Journal, 292, 371

Dehnen W., 2014, Computational Astrophysics and Cosmology, 1, 1

Dubois Y., Devriendt J., Slyz A., Teyssier R., 2010, Monthly Notices of the Royal Astronomical Society, 409, 985

Eisenstein D. J., Hut P., 1998, The Astrophysical Journal, 498, 137

Federrath C., Banerjee R., Clark P. C., Klessen R. S., 2010, The Astrophysical Journal, 713, 269

Federrath C., Sur S., Schleicher D. R., Banerjee R., Klessen R. S., 2011, The Astrophysical Journal, 731, 62

Fromang S., Hennebelle P., Teyssier R., 2006, Astronomy and Astrophysics, 457, 371

Fryxell B. et al., 2000, Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes

Gingold R. A., Monaghan J. J., 1977, Monthly Notices of the Royal Astronomical Society, 181, 375

Girichidis P., Federrath C., Banerjee R., Klessen R. S., 2011, Monthly Notices of the Royal Astronomical Society, 413, 2741

Godunov S. K., 1959, Matematicheskii Sbornik, 89, 271

Gong H., Ostriker E. C., 2013, The Astrophysical Journal, 204, 8

Goodwin S. P., Whitworth A. P., Ward-Thompson D., 2004, Astronomy and Astrophysics, 414, 633

Guillet T., Teyssier R., 2011, Journal of Computational Physics, 230, 4756

Heitsch F., Hartmann L. W., Slyz A. D., Devriendt J. E. G., Burkert A., 2008, The Astrophysical Journal, 674, 316

Hennebelle P., Chabrier G., 2008, The Astrophysical Journal, 684, 395

Hennebelle P., Commerçon B., Joos M., Klessen R., Krumholz M., Tan J., Teyssier R., 2011, Astronomy and Astrophysics, 528, A72

Hennebelle P., Falgarone E., 2012, The Astronomy and Astrophysics Review, 20, 1

Heyer M., Krawczyk C., Duval J., Jackson J. M., 2009, The Astrophysical Journal, 699, 1092

Hockney R. W., Eastwood J. W., 1981, Computer Simulation Using Particles

Hopkins P. F., 2012, Monthly Notices of the Royal Astronomical Society, 423, 2037

Hoyle F., Lyttleton R. A., 1939, Proceedings of the Cambridge Philosophical Society, 35, 405

Hubber D. A., Batty C. P., McLeod A., Whitworth A. P., 2011, Astronomy and Astrophysics, 529, A27

Hubber D. A., Walch S., Whitworth A. P., 2013, Monthly Notices of the Royal Astronomical Society, 430, 3261

Hubber D. A., Whitworth A. P., 2005, Astronomy and Astrophysics, 437, 113

Jappsen A.-K., Klessen R. S., Larson R. B., Li Y., Mac Low M.-M., 2005, Astronomy and Astrophysics, 435, 611

Jeans J. H., 1902, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1

Khokhlov A. M., 1998, Journal of Computational Physics, 143, 519

Kitsionas S., Whitworth A., 2002, Monthly Notices of the Royal Astronomical Society, 330, 129

Klessen R. S., 2011, EAS Publications Series, 51, 133

Klessen R. S., Ballesteros-Paredes J., Li Y., Mac Low M.-M., 2004, in Astronomical Society of the Pacific Conference Series, Vol. 322, The Formation and Evolution of Massive Young Star Clusters, Lamers H. J. G. L. M., Smith L. J., Nota A., eds., pp. 299–308

Knebe A. et al., 2011, Monthly Notices of the Royal Astronomical Society, 415, 2293

Knebe A. et al., 2013, Monthly Notices of the Royal Astronomical Society, 435, 1618

Knollmann S. R., Knebe A., 2009, The Astrophysical Journal Supplement Series, 182, 608

Knuth D. E., 1998, The Art of Computer Programming, Vol. 3, Sorting and Searching. Addison-Wesley, Reading, Massachusetts

Kolmogorov A. N., 1941a, in Dokl. Akad. Nauk SSSR, Vol. 32, pp. 19–21

Kolmogorov A. N., 1941b, in Dokl. Akad. Nauk SSSR, Vol. 30, pp. 299–303

Kritsuk A. G., Lee C. T., Norman M. L., 2013, Monthly Notices of the Royal Astronomical Society, stt1805

Kritsuk A. G., Norman M. L., Padoan P., Wagner R., 2007, The Astrophysical Journal, 665, 416

Kroupa P., 2001, Monthly Notices of the Royal Astronomical Society, 322, 231

Kroupa P., 2002, Science, 295, 82

Krumholz M. R., 2014, Physics Reports, 539, 49

Krumholz M. R., Klein R. I., McKee C. F., 2007, The Astrophysical Journal, 656, 959

Krumholz M. R., Klein R. I., McKee C. F., 2011, The Astrophysical Journal, 740, 74

Krumholz M. R., Klein R. I., McKee C. F., 2012, The Astrophysical Journal, 754, 71

Krumholz M. R., McKee C. F., Klein R. I., 2004, The Astrophysical Journal, 611, 399

Lada C. J., Lada E. A., 2003, Annual Review of Astronomy and Astrophysics, 41, 57

Larson R. B., 1969, Monthly Notices of the Royal Astronomical Society, 145, 271

Larson R. B., 1981, Monthly Notices of the Royal Astronomical Society, 194, 809

Larson R. B., 2005, Monthly Notices of the Royal Astronomical Society, 359, 211

Lee A. T., Cunningham A. J., McKee C. F., Klein R. I., 2014, The Astrophysical Journal, 783, 50

LeVeque R. J., 2002, Finite volume methods for hyperbolic problems, Vol. 31. Cambridge university press

Levermore C. D., 1984, Journal of Quantitative Spectroscopy and Radiative Transfer, 31, 149

Liebendörfer M., Whitehouse S., Fischer T., 2009, The Astrophysical Journal, 698, 1174

Lucy L., 1977, The Astronomical Journal, 82, 1013

Martel H., Evans II N. J., Shapiro P. R., 2006, The Astrophysical Journal Supplement Series, 163, 122

Masunaga H., Inutsuka S.-i., 2000, The Astrophysical Journal, 531, 350

Masunaga H., Miyama S. M., Inutsuka S.-i., 1998, The Astrophysical Journal, 495, 346

Meyer F., 1994, Signal processing, 38, 113

Minerbo G. N., 1978, Journal of Quantitative Spectroscopy and Radiative Transfer, 20, 541

Moga A., 1997, Parallel watershed algorithms for image segmentation. Tampere University of Technology

Moga A. N., Gabbouj M., 1998, Journal of Parallel and Distributed Computing, 51, 27

Myers A. T., Krumholz M. R., Klein R. I., McKee C. F., 2011, The Astrophysical Journal, 735, 49

Myers A. T., McKee C. F., Cunningham A. J., Klein R. I., Krumholz M. R., 2013, The Astrophysical Journal, 766, 97

Ocvirk P., 2014, in Exascale Radio Astronomy, Vol. 1, p. 40401

Offner S. S., Clark P. C., Hennebelle P., Bastian N., Bate M. R., Hopkins P. F., Moraux E., Whitworth A. P., 2013, arXiv preprint arXiv:1312.5326

Offner S. S. R., Klein R. I., McKee C. F., Krumholz M. R., 2009, The Astrophysical Journal, 703, 131

Onions J. et al., 2013, Monthly Notices of the Royal Astronomical Society, 429, 2739

O'Shea B. W., Bryan G., Bordner J., Norman M. L., Abel T., Harkness R., Kritsuk A., 2004, ArXiv Astrophysics e-prints

Padoan P., Haugbølle T., Nordlund Å., 2014, The Astrophysical Journal, 797, 32

Padoan P., Nordlund Å., 2011, The Astrophysical Journal, 730, 40

Padoan P., Nordlund Å., Kritsuk A. G., Norman M. L., Li P. S., 2007, The Astrophysical Journal, 661, 972

Passy J.-C., Bryan G. L., 2014, The Astrophysical Journal Supplement Series, 215, 8

Peng B., Zhang D., 2011, Image Processing, IEEE Transactions on, 20, 3592

Peters T., Banerjee R., Klessen R. S., Mac Low M.-M., 2011, The Astrophysical Journal, 729, 72

Peters T., Mac Low M.-M., Banerjee R., Klessen R. S., Dullemond C. P., 2010, The Astrophysical Journal, 719, 831

Press W. H., Schechter P., 1974, The Astrophysical Journal, 187, 425

Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 2007, Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd edn. Cambridge University Press, New York, NY, USA

Pujol A. et al., 2014, Monthly Notices of the Royal Astronomical Society, 438, 3205

Roerdink J. B., Meijster A., 2000, Fundam. Inf., 41, 187

Roman-Duval J., Jackson J. M., Heyer M., Rathborne J., Simon R., 2010, The Astrophysical Journal, 723, 492

Rosdahl J., Blaizot J., Aubert D., Stranex T., Teyssier R., 2013, Monthly Notices of the Royal Astronomical Society, 436, 2188

Rosolowsky E., Pineda J., Kauffmann J., Goodman A., 2008, The Astrophysical Journal, 679, 1338

Salpeter E. E., 1955, The Astrophysical Journal, 121, 161

Schmidt W., Kern S. A. W., Federrath C., Klessen R. S., 2010, Astronomy and Astrophysics, 516, A25

Semenov D., Henning T., Helling C., Ilgner M., Sedlmayr E., 2003, Astronomy and Astrophysics, 410, 611

Shen S., Wadsley J., 2006, The Astrophysical Journal, 651, L145

Shu F. H., 1977, The Astrophysical Journal, 214, 488

Shu F. H., 1991, The Physics of Astrophysics: Gas Dynamics, Vol. 2. University Science Books

Skory S., Turk M. J., Norman M. L., Coil A. L., 2010, The Astrophysical Journal Supplement Series, 191, 43

Smith R. J., Clark P. C., Bonnell I. A., 2009, Monthly Notices of the Royal Astronomical Society, 396, 830

Springel V., White S. D. M., Tormen G., Kauffmann G., 2001, Monthly Notices of the Royal Astronomical Society, 328, 726

Stadel J. G., 2001, PhD thesis, UNIVERSITY OF WASHINGTON

Stahler S. W., Palla F., 2008, The formation of stars. John Wiley & Sons

Stamatellos D., Whitworth A., Bisbas T., Goodwin S., 2007, Astronomy and Astrophysics, 475, 37

Stamatellos D., Whitworth A. P., Hubber D. A., 2011, The Astrophysical Journal, 730, 32

Stellingwerf R. F., Buff J., 1978, The Astrophysical Journal, 221, 661

Stutzki J., Guesten R., 1990, The Astrophysical Journal, 356, 513

Teyssier R., 2002, Astronomy and Astrophysics, 385, 337

Toro E. F., 2009, Riemann solvers and numerical methods for fluid dynamics : a practical introduction. Springer, New York

Truelove J. K., Klein R. I., McKee C. F., Holliman, II J. H., Howell L. H., Greenough J. A., 1997, The Astrophysical Journal, 489, L179

van Leer B., 1979, Journal of Computational Physics, 32, 101

Wadsley J., Reid M., Qamar F., Sills A., Petitclerc N., 2011, in IAU Symposium, Vol. 270, Computational Star Formation, Alves J., Elmegreen B. G., Girart J. M., Trimble V., eds., pp. 129–132

Walch S., Whitworth A., Bisbas T., Wünsch R., Hubber D., 2013, Monthly Notices of the Royal Astronomical Society, 435, 917

Walch S., Whitworth A. P., Girichidis P., 2012, Monthly Notices of the Royal Astronomical Society, 419, 760

Wang P., Li Z.-Y., Abel T., Nakamura F., 2010, The Astrophysical Journal, 709, 27

Ward-Thompson D., Whitworth A. P., Whitworth A. P., 2011, An introduction to star formation. Cambridge University Press

Williams J. P., De Geus E. J., Blitz L., 1994, The Astrophysical Journal, 428, 693